

Please read the following information regarding using the meter APIs carefully.

Classes:

**PinReader:** Reads data from pin.

methods:

**PinReader():** Constructor

**read():** Returns the current value being transmitted to the pin from the sensors (for example voltage).

**Communicator:** provides functionality for communicating with the server.

methods:

**Communicator(String host, int port):** Constructor. Takes host address and port number of the server. If you pass 0 as the port number, the default port number 5555 will be used. If you return "" as host, the default host "localhost" will be used. If you are assuming the server is on the same machine, use "localhost" as host.

**public void sendMessage(String message):** sends \*message\* to the server.

**public String recvMessage():** Returns any message that has been received from the server and is in the receiver buffer.

**public void close():** Closes the communication with the server.

Using PinReader class:

- 1- Put PinReader.class in ./meterAPI folder (Create the folder meterAPI inside the directory where your codes reside)
- 2- put "import meterAPI.\*" in your code
- 3- Instantiate an object of the type PinReader:  
example: `PinReader pin = new PinReader();`
- 4- To read data from pin, call read() method. It will return the current data transmitted to the pin.

**\*\* Note:** read() does not calculate consumption. You have to calculate it yourself. It only returns the current value (voltage for instance) on the wire.

Using Communicator class:

- 1- Put Communicator.class in ./meterAPI folder (Create the folder meterAPI inside the directory where your codes reside)
- 2- put "import meterAPI.\*" in your code
- 3- Define an object of the type Communicator:  
example: `Communicator comm = new Communicator("localhost", 5555);`
- 4- To send data to the server, use sendMessage(String message) method.  
example: `comm.sendMessage("This is my data");`
- 5- To check if any data is received from the server call recvMessage() method:  
example: `String inputData = comm.recvMessage();`

**\*\*\*\*** Please have a look at testServer.java and testMeter.java for a simple example.