EPANET Simulation with Modbus Controls

This Python script runs an EPANET simulation with Modbus controls, enabling real-time interaction between the EPANET model/network and external PLCs. The script reads control values from PLCs via Modbus, applies them to the EPANET simulation, and then writes the simulation results back to the PLCs.

To achieve this, we have minimized the use of external modules, keeping the script as close as possible to default Python functionalities. The only modules we used are:

- Python standard library: `sys`, `time`
- Third-party libraries: `epyt`, `pymodbus`

The script consists of the following functions:

- `def parse_arguments() -> str:`
- `def setup_epanet(inp_file: str) -> epanet:`
- `def get_zones(en: epanet) -> set[str]:`
- `def setup_clients(zones: set) -> dict[str, ModbusTcpClient]:`
- `def get_controls(clients: dict[str, ModbusTcpClient], en: epanet) -> dict:`
- `def set_controls(en: epanet, controls: dict) -> None:`
- `def read_data(en: epanet) -> dict:`
- `def write_data(clients: dict[str, ModbusTcpClient], data: dict) -> None:`
- `def main():`

In the section below, we'll explain each function in depth and how they work under the hood.

- `def parse_arguments() -> str:`

The `parse_arguments` function is self-explanatory and has been designed for simplicity. It handles command-line arguments to ensure a valid `.inp` file is provided.

Function:

```
def parse_arguments() -> str:
    if len(sys.argv) != 2 or not sys.argv[1].endswith(".inp"):
        print("Run EPANET simulation with Modbus controls.")
        print(f">>> python {sys.argv[0]} [network.inp]")
        sys.exit(1)
    return sys.argv[1]
```

If a valid EPANET network is provided it will return a string containing the path to the `.inp` file. Else if no valid `.inp` file is provided, the program exits and displays a message showing how to use the script.

- `def setup_epanet(inp_file: str) -> epanet:`

The `setup_epanet` function ...

Function:

```
def setup_epanet(inp_file: str) -> epanet:
    try:
        en: epanet = epanet(inp_file)
        en.setTimeSimulationDuration(10)  # initial setup; duration will be set to infinite in main
function.
        en.setTimeHydraulicStep(1)
        return en
    except Exception as e:
        print(f"ERROR in setup_epanet: {e}")
        sys.exit(1)
```

Something here...

- `def get_zones(en: epanet) -> set[str]:`

The `get_zones` function ...

Function:

```
def get_zones(en: epanet) -> set[str]:
    try:
        zones: set[str] = set()

        for name_id in en.getNodeNameID() + en.getLinkNameID():
            if "-" not in name_id:
                continue
            zone, _ = name_id.split("-", 1)
            zones.add(zone)
        return zones
    except Exception as e:
        print(f"ERROR in get_zones: {e}")
        sys.exit(1)
```

Something here...

- `def setup_clients(zones: set) -> dict[str, ModbusTcpClient]:`

The `setup_clients` function ... https://github.com/Ketho/OT-Simulation/blob/test/docs/openplc.md#adding-a-plc

Function:

```
def setup_clients(zones: set) -> dict[str, ModbusTcpClient]:
    try:
        clients: dict[str, ModbusTcpClient] = {
            zone: ModbusTcpClient(host=f'plc-{zone}', port=502)
            for zone in zones
        }
        # clients: dict[str, ModbusTcpClient] = {
        #     zone: ModbusTcpClient(host="127.0.0.1", port=502 + i)
        #     for i, zone in enumerate(zones)
        # }
        for _, client in clients.items():
            while not client.connect():
                time.sleep(1)
        return clients
    except Exception as e:
        print(f"ERROR in setup_clients: {e}")
        sys.exit(1)
```

Something here...

- `def get_controls(clients: dict[str, ModbusTcpClient], en: epanet) -> dict:`

The `get_controls` function ...

Function:

```
def get_controls(clients: dict[str, ModbusTcpClient], en: epanet) -> dict:
    try:
        controls: dict = {}

        for name_id in en.getNodeNameID() + en.getLinkNameID():
            if "-" not in name_id:
                continue
            zone, element = name_id.split("-", 1)
            controls.setdefault(zone, {})

            if name_id in en.getLinkNameID():
                link_index: int = en.getLinkIndex(name_id)

                match en.getLinkType(link_index):
                    case "PIPE":
                        pass
                    case "PUMP":
                        controls[zone].setdefault(element, {})
                        controls[zone][element]["speed"] = None
                    case _:
                        controls[zone].setdefault(element, {})
                        controls[zone][element]["setting"] = None

        for zone, client in clients.items():
            pump_count = sum(1 for element in controls[zone] if "speed" in controls[zone][element])

            if pump_count > 0:
                pump_registers = client.read_holding_registers(address=1000, count=pump_count *
2).registers  # this function caused a weird error on the server only, but defining the function
parameters this way, the error was solved :)

            for i, element in enumerate(e for e in controls[zone] if "speed" in controls[zone][e]):
                converted_value = client.convert_from_registers(
                    pump_registers[i * 2 : i * 2 + 2], client.DATATYPE.FLOAT32
                )
                controls[zone][element]["speed"] = converted_value

            valve_count = sum(1 for element in controls[zone] if "setting" in controls[zone][element])

            if valve_count > 0:
                valve_registers = client.read_holding_registers(address=2000, count=valve_count *
2).registers  # same applies here...

            for i, element in enumerate(e for e in controls[zone] if "setting" in controls[zone][e]):
                converted_value = client.convert_from_registers(
                    valve_registers[i * 2 : i * 2 + 2], client.DATATYPE.FLOAT32
                )
                controls[zone][element]["setting"] = converted_value

        return controls
    except Exception as e:
        print(f"Error in get_controls: {e}")
        sys.exit(1)
```

```
{'zone0': {'valve1': {'setting': 1.8367379491291107e-40}, 'valve2': {'setting': 1.8367379491291107e-40},
'valve3': {'setting': 1.8367379491291107e-40}, 'valve4': {'setting': 1.8367379491291107e-40}}, 'zone3':
{'pump1': {'speed': 9.183689745645554e-41}, 'pump2': {'speed': 9.183689745645554e-41}, 'pump3': {'speed':
9.183689745645554e-41}}, 'zone1': {'pump1': {'speed': 9.183689745645554e-41}, 'pump2': {'speed':
9.183689745645554e-41}, 'pump3': {'speed': 9.183689745645554e-41}}, 'zone4': {'pump2': {'speed':
9.183689745645554e-41}, 'pump3': {'speed': 9.183689745645554e-41}, 'pump1': {'speed': 9.183689745645554e-
41}}, 'zone2': {'pump2': {'speed': 9.183689745645554e-41}, 'pump3': {'speed': 9.183689745645554e-41},
'pump1': {'speed': 9.183689745645554e-41}}}
```

```
{
    "zone0": {
        "valve1": {
            "setting": 1.8367379491291107e-40
        },
        "valve2": {
            "setting": 1.8367379491291107e-40
        },
        "valve3": {
            "setting": 1.8367379491291107e-40
        },
        "valve4": {
            "setting": 1.8367379491291107e-40
        }
    },
    "zone3": {
        "pump1": {
            "speed": 9.183689745645554e-41
        },
        "pump2": {
            "speed": 9.183689745645554e-41
        },
        "pump3": {
            "speed": 9.183689745645554e-41
        }
    },
    "zone1": {
        "pump1": {
            "speed": 9.183689745645554e-41
        },
        "pump2": {
            "speed": 9.183689745645554e-41
        },
        "pump3": {
            "speed": 9.183689745645554e-41
        }
    },
    "zone4": {
        "pump2": {
            "speed": 9.183689745645554e-41
        },
        "pump3": {
            "speed": 9.183689745645554e-41
        },
        "pump1": {
            "speed": 9.183689745645554e-41
        }
    },
    "zone2": {
        "pump2": {
            "speed": 9.183689745645554e-41
        },
        "pump3": {
            "speed": 9.183689745645554e-41
        },
        "pump1": {
            "speed": 9.183689745645554e-41
        }
    }
}
```

- `def set_controls(en: epanet, controls: dict) -> None:`

The `set_controls` function ...

Function:

```
def set_controls(en: epanet, controls: dict) -> None:
    try:
        # offset_speed = 1000
        # offset_setting = 2000

        for zone, elements in controls.items():
            for element, control in elements.items():
                link_index: int = en.getLinkIndex(f"{zone}-{element}")

                if "speed" in control:
                    en.setLinkSettings(link_index, control["speed"])
                    # print(f"{zone:<15} -> {element:<15} -> register: {offset_speed:<15} control:
speed")
                    # offset_speed += 2

                if "setting" in control:
                    en.setLinkSettings(link_index, control["setting"])
                    # print(f"{zone:<15} -> {element:<15} -> register: {offset_setting:<15} control:
setting")
                    # offset_setting += 2

            # print()
    except Exception as e:
        print(f"ERROR in set_controls: {e}")
        sys.exit(1)
```

Something here...

```
zone0           -> valve1         -> register: 2000           control: setting
zone0           -> valve2         -> register: 2002           control: setting
zone0           -> valve3         -> register: 2004           control: setting
zone0           -> valve4         -> register: 2006           control: setting

zone3           -> pump1          -> register: 1000           control: speed
zone3           -> pump2          -> register: 1002           control: speed
zone3           -> pump3          -> register: 1004           control: speed

zone1           -> pump1          -> register: 1006           control: speed
zone1           -> pump2          -> register: 1008           control: speed
zone1           -> pump3          -> register: 1010           control: speed

zone4           -> pump2          -> register: 1012           control: speed
zone4           -> pump3          -> register: 1014           control: speed
zone4           -> pump1          -> register: 1016           control: speed

zone2           -> pump2          -> register: 1018           control: speed
zone2           -> pump3          -> register: 1020           control: speed
zone2           -> pump1          -> register: 1022           control: speed
```

- `def read_data(en: epanet) -> dict:`

The `read_data` function ...

Function:

```
def read_data(en: epanet) -> dict:
    try:
        data: dict = {}

        for name_id in en.getNodeNameID() + en.getLinkNameID():
            if "-" not in name_id:
                continue
            zone, element = name_id.split("-", 1)
            data.setdefault(zone, {}).setdefault(element, {})

            e: dict = data[zone][element]

            if name_id in en.getNodeNameID():
                node_index: int = en.getNodeIndex(name_id)

                e["index"] = str(node_index)
                e["hydraulic_head"] = str(en.getNodeHydraulicHead(node_index))
                e["pressure"] = str(en.getNodePressure(node_index))
                e["elevation"] = str(en.getNodeElevations(node_index))

                if en.getNodeType(node_index) == "TANK":
                    e["minimum_water_level"] = str(en.getNodeTankMinimumWaterLevel(node_index))
                    e["maximum_water_level"] = str(en.getNodeTankMaximumWaterLevel(node_index))
                    e["initial_water_level"] = str(en.getNodeTankInitialLevel(node_index))
                    e["minimum_water_volume"] = str(en.getNodeTankMinimumWaterVolume(node_index))
                    e["maximum_water_volume"] = str(en.getNodeTankMaximumWaterVolume(node_index))
                    e["initial_water_volume"] = str(en.getNodeTankInitialWaterVolume(node_index))

            if name_id in en.getLinkNameID():
                link_index: int = en.getLinkIndex(name_id)

                e["index"] = str(link_index)
                e["status"] = str(en.getLinkStatus(link_index))
                e["flow_rate"] = str(en.getLinkFlows(link_index))

                match en.getLinkType(link_index):
                    case "PIPE":
                        pass
                    case "PUMP":
                        e["power"] = str(en.getLinkPumpPower(link_index))
                        e["speed"] = str(en.getLinkSettings(link_index))
                        e["energy_usage"] = str(en.getLinkEnergy(link_index))
                    case _:  # default case to handle all valve types.
                        e["setting"] = str(en.getLinkSettings(link_index))

        return data
    except Exception as e:
        print(f"ERROR in read_data: {e}")
        sys.exit(1)
```

```
{'zone0': {'junction2': {'index': '1', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction3': {'index': '2', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction4': {'index': '3', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction5': {'index': '4', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction6': {'index': '5', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction1': {'index': '147', 'hydraulic_head': '1.99999996490334e-13', 'pressure': '1.99999996490334e-
13', 'elevation': '0.0'}, 'reservoir1': {'index': '149', 'hydraulic_head': '0.0', 'pressure': '0.0',
'elevation': '0.0'}, 'valve1': {'index': '152', 'status': '1', 'flow_rate': '0.0904926210641861',
'setting': '0.0'}, 'valve2': {'index': '153', 'status': '1', 'flow_rate': '0.0', 'setting': '0.0'},
'valve3': {'index': '154', 'status': '1', 'flow_rate': '-0.0005311733111739159', 'setting': '0.0'},
'valve4': {'index': '155', 'status': '1', 'flow_rate': '0.0005311733111739159', 'setting': '0.0'}},
'zone3': {'junction1': {'index': '6', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction6': {'index': '34', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction2': {'index': '35', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction5': {'index': '36', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction3': {'index': '37', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'junction4': {'index': '38', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node40':
{'index': '39', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node39': {'index':
'40', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node38': {'index': '41',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node37': {'index': '42',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node36': {'index': '43',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node35': {'index': '44',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node34': {'index': '45',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node33': {'index': '46',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node32': {'index': '47',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node31': {'index': '48',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node41': {'index': '49',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node1': {'index': '50',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node2': {'index': '51',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node3': {'index': '52',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node4': {'index': '53',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node5': {'index': '54',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node6': {'index': '55',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node7': {'index': '56',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node29': {'index': '57',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node30': {'index': '58',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node23': {'index': '59',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node24': {'index': '60',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node25': {'index': '61',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node26': {'index': '62',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node27': {'index': '63',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node28': {'index': '64',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node8': {'index': '65',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node18': {'index': '66',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node17': {'index': '67',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node16': {'index': '68',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node15': {'index': '69',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node14': {'index': '70',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node13': {'index': '71',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node12': {'index': '72',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node11': {'index': '73',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node10': {'index': '74',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node9': {'index': '75',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node22': {'index': '76',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node21': {'index': '77',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node20': {'index': '78',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node19': {'index': '79',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'tank1': {'index': '153',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0', 'minimum_water_level': '0.0',
'maximum_water_level': '50.0', 'initial_water_level': '50.0', 'minimum_water_volume': '0.0',
'maximum_water_volume': '98174.7734375', 'initial_water_volume': '98174.7734375'}, 'pump1': {'index':
'141', 'status': '0', 'flow_rate': '0.0', 'power': '1.0', 'speed': '1.0193895617666565e-38',
'energy_usage': '0.0'}, 'pump2': {'index': '144', 'status': '1', 'flow_rate': '0.02326112985610962',
'power': '1.0', 'speed': '1.0193895617666565e-38', 'energy_usage': '2.0630763356473348e-18'}, 'pump3':
{'index': '147', 'status': '1', 'flow_rate': '0.0004578706284519285', 'power': '1.0', 'speed':
'1.0193895617666565e-38', 'energy_usage': '0.0'}}, 'zone1': {'junction2': {'index': '7',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction3': {'index': '8',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction4': {'index': '9',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction1': {'index': '10',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node2': {'index': '11',
'hydraulic_head': '49.998661041259766', 'pressure': '49.998661041259766', 'elevation': '0.0'}, 'node3':
{'index': '12', 'hydraulic_head': '49.998661041259766', 'pressure': '49.998661041259766', 'elevation':
'0.0'}, 'node4': {'index': '13', 'hydraulic_head': '49.998661041259766', 'pressure':
'49.998661041259766', 'elevation': '0.0'}, 'node5': {'index': '14', 'hydraulic_head':
'49.998661041259766', 'pressure': '49.998661041259766', 'elevation': '0.0'}, 'node6': {'index': '15',
'hydraulic_head': '49.998661041259766', 'pressure': '49.998661041259766', 'elevation': '0.0'}, 'node7':
{'index': '16', 'hydraulic_head': '49.998661041259766', 'pressure': '49.998661041259766', 'elevation':
'0.0'}, 'node11': {'index': '17', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'node12': {'index': '18', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node13':
{'index': '19', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node14': {'index':
'20', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node15': {'index': '21',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node16': {'index': '22',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node10': {'index': '23',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node8': {'index': '24',
'hydraulic_head': '49.998661041259766', 'pressure': '49.998661041259766', 'elevation': '0.0'}, 'node9':
{'index': '25', 'hydraulic_head': '49.998661041259766', 'pressure': '49.998661041259766', 'elevation':
'0.0'}, 'node1': {'index': '26', 'hydraulic_head': '49.998661041259766', 'pressure':
'49.998661041259766', 'elevation': '0.0'}, 'node18': {'index': '27', 'hydraulic_head': '50.0',
'pressure': '50.0', 'elevation': '0.0'}, 'node19': {'index': '28', 'hydraulic_head': '50.0', 'pressure':
'50.0', 'elevation': '0.0'}, 'node20': {'index': '29', 'hydraulic_head': '50.0', 'pressure': '50.0',
'elevation': '0.0'}, 'node21': {'index': '30', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation':
'0.0'}, 'node22': {'index': '31', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'},
'node23': {'index': '32', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node24':
{'index': '33', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node17': {'index':
'148', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'tank1': {'index': '150',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0', 'minimum_water_level': '0.0',
'maximum_water_level': '50.0', 'initial_water_level': '50.0', 'minimum_water_volume': '0.0',
'maximum_water_volume': '98174.7734375', 'initial_water_volume': '98174.7734375'}, 'pump1': {'index':
'140', 'status': '0', 'flow_rate': '0.0', 'power': '1.0', 'speed': '1.0193895617666565e-38',
'energy_usage': '0.0'}, 'pump2': {'index': '142', 'status': '1', 'flow_rate': '0.02326112985610962',
'power': '1.0', 'speed': '1.0193895617666565e-38', 'energy_usage': '2.0630763356473348e-18'}, 'pump3':
{'index': '146', 'status': '1', 'flow_rate': '0.09038212150335312', 'power': '1.0', 'speed':
'1.0193895617666565e-38', 'energy_usage': '3.138246150793097e-17'}}, 'zone4': {'junction1': {'index':
'80', 'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction7': {'index': '81',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction2': {'index': '82',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction3': {'index': '83',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction6': {'index': '84',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction4': {'index': '85',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction5': {'index': '86',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node31': {'index': '87',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node32': {'index': '88',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node35': {'index': '89',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node34': {'index': '90',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node33': {'index': '91',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node11': {'index': '92',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node10': {'index': '93',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node12': {'index': '94',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node6': {'index': '95',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node7': {'index': '96',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node8': {'index': '97',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node9': {'index': '98',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node1': {'index': '99',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node2': {'index': '100',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node4': {'index': '101',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node5': {'index': '102',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node3': {'index': '103',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node15': {'index': '104',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node14': {'index': '105',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node13': {'index': '106',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node20': {'index': '107',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node19': {'index': '108',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node18': {'index': '109',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node17': {'index': '110',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node16': {'index': '111',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node24': {'index': '112',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node23': {'index': '113',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node21': {'index': '114',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node22': {'index': '115',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node28': {'index': '116',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node27': {'index': '117',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node29': {'index': '118',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node30': {'index': '119',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node25': {'index': '120',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node26': {'index': '121',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'tank1': {'index': '152',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0', 'minimum_water_level': '0.0',
'maximum_water_level': '50.0', 'initial_water_level': '50.0', 'minimum_water_volume': '0.0',
'maximum_water_volume': '98174.7734375', 'initial_water_volume': '98174.7734375'}, 'pump2': {'index':
'145', 'status': '1', 'flow_rate': '0.02326112985610962', 'power': '1.0', 'speed': '1.0193895617666565e-
38', 'energy_usage': '2.0630763356473348e-18'}, 'pump3': {'index': '148', 'status': '1', 'flow_rate':
'8.897751831682399e-05', 'power': '1.0', 'speed': '1.0193895617666565e-38', 'energy_usage':
'5.037188637665904e-22'}, 'pump1': {'index': '151', 'status': '0', 'flow_rate': '0.0', 'power': '1.0',
'speed': '1.0193895617666565e-38', 'energy_usage': '0.0'}}, 'zone2': {'junction1': {'index': '122',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction3': {'index': '123',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction4': {'index': '124',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'junction2': {'index': '125',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node1': {'index': '126',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node2': {'index': '127',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node3': {'index': '128',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node4': {'index': '129',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node5': {'index': '130',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node6': {'index': '131',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node11': {'index': '132',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node12': {'index': '133',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node13': {'index': '134',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node14': {'index': '135',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node10': {'index': '136',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node17': {'index': '137',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node15': {'index': '138',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node16': {'index': '139',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node18': {'index': '140',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node19': {'index': '141',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node20': {'index': '142',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node21': {'index': '143',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node8': {'index': '144',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node9': {'index': '145',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'node7': {'index': '146',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0'}, 'tank1': {'index': '151',
'hydraulic_head': '50.0', 'pressure': '50.0', 'elevation': '0.0', 'minimum_water_level': '0.0',
'maximum_water_level': '50.0', 'initial_water_level': '50.0', 'minimum_water_volume': '0.0',
'maximum_water_volume': '98174.7734375', 'initial_water_volume': '98174.7734375'}, 'pump2': {'index':
'143', 'status': '1', 'flow_rate': '0.02326112985610962', 'power': '1.0', 'speed': '1.0193895617666565e-
38', 'energy_usage': '2.0630763356473348e-18'}, 'pump3': {'index': '149', 'status': '1', 'flow_rate':
'0.00035591007326729596', 'power': '1.0', 'speed': '1.0193895617666565e-38', 'energy_usage': '0.0'},
'pump1': {'index': '150', 'status': '0', 'flow_rate': '0.0', 'power': '1.0', 'speed':
'1.0193895617666565e-38', 'energy_usage': '0.0'}}}
```

```
[---TRUNCATED---]
        }
    },
    "zone2": {
        "junction1": {
            "index": "122",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "junction3": {
            "index": "123",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "junction4": {
            "index": "124",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "junction2": {
            "index": "125",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node1": {
            "index": "126",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node2": {
            "index": "127",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node3": {
            "index": "128",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node4": {
            "index": "129",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node5": {
            "index": "130",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node6": {
            "index": "131",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node11": {
            "index": "132",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node12": {
            "index": "133",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node13": {
            "index": "134",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node14": {
            "index": "135",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node10": {
            "index": "136",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node17": {
            "index": "137",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node15": {
            "index": "138",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node16": {
            "index": "139",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node18": {
            "index": "140",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node19": {
            "index": "141",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node20": {
            "index": "142",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node21": {
            "index": "143",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node8": {
            "index": "144",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node9": {
            "index": "145",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "node7": {
            "index": "146",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0"
        },
        "tank1": {
            "index": "151",
            "hydraulic_head": "50.0",
            "pressure": "50.0",
            "elevation": "0.0",
            "minimum_water_level": "0.0",
            "maximum_water_level": "50.0",
            "initial_water_level": "50.0",
            "minimum_water_volume": "0.0",
            "maximum_water_volume": "98174.7734375",
            "initial_water_volume": "98174.7734375"
        },
        "pump2": {
            "index": "143",
            "status": "1",
            "flow_rate": "0.02326112985610962",
            "power": "1.0",
            "speed": "1.0193895617666565e-38",
            "energy_usage": "2.0630763356473348e-18"
        },
        "pump3": {
            "index": "149",
            "status": "1",
            "flow_rate": "0.00035591007326729596",
            "power": "1.0",
            "speed": "1.0193895617666565e-38",
            "energy_usage": "0.0"
        },
        "pump1": {
            "index": "150",
            "status": "0",
            "flow_rate": "0.0",
            "power": "1.0",
            "speed": "1.0193895617666565e-38",
            "energy_usage": "0.0"
        }
    }
}
```

- `def write_data(clients: dict[str, ModbusTcpClient], data: dict) -> None:`

The `write_data` function ...

Function:

```python
def write_data(clients: dict[str, ModbusTcpClient], data: dict) -> None:
    try:
        for zone, elements in data.items():
            client: ModbusTcpClient = clients[zone]
            offset: int = 0

            for element, values in elements.items():
                for i, (k, value) in enumerate(values.items()):
                    address: int = offset + i * 2
                    registers: list[int] = client.convert_to_registers(
                        float(value), client.DATATYPE.FLOAT32
                    )

                    client.write_registers(address, registers)

                    # print(
                    #     f"{zone:<15} -> {element:<15} -> {k:<30}: {value:<30}, "
                    #     f"registers: {str(registers):<20}, address: {address}"
                    # )

                offset += len(values) * 2

                # print()  # blank lines for separating log entries.
            # print()
            # print()
    except Exception as e:
        print(f"ERROR in write_data: {e}")
        sys.exit(1)
```

Something here...

Snippet...

```
[---TRUNCATED---]
zone3           -> junction1       -> index                          : 6                             ,
registers: [16576, 0]          , address: 0
zone3           -> junction1       -> hydraulic_head                 : 514.2538452148438             ,
registers: [17408, 36927]      , address: 2
zone3           -> junction1       -> pressure                       : 514.2538452148438             ,
registers: [17408, 36927]      , address: 4
zone3           -> junction1       -> elevation                      : 0.0                           ,
registers: [0, 0]              , address: 6

zone3           -> junction6       -> index                          : 34                            ,
registers: [16904, 0]          , address: 8
zone3           -> junction6       -> hydraulic_head                 : 514.2538452148438             ,
registers: [17408, 36927]      , address: 10
zone3           -> junction6       -> pressure                       : 514.2538452148438             ,
registers: [17408, 36927]      , address: 12
zone3           -> junction6       -> elevation                      : 0.0                           ,
registers: [0, 0]              , address: 14

zone3           -> junction2       -> index                          : 35                            ,
registers: [16908, 0]          , address: 16
zone3           -> junction2       -> hydraulic_head                 : 514.2538452148438             ,
registers: [17408, 36927]      , address: 18
zone3           -> junction2       -> pressure                       : 514.2538452148438             ,
registers: [17408, 36927]      , address: 20
zone3           -> junction2       -> elevation                      : 0.0                           ,
registers: [0, 0]              , address: 22

zone3           -> junction5       -> index                          : 36                            ,
registers: [16912, 0]          , address: 24
zone3           -> junction5       -> hydraulic_head                 : 514.2538452148438             ,
registers: [17408, 36927]      , address: 26
zone3           -> junction5       -> pressure                       : 514.2538452148438             ,
registers: [17408, 36927]      , address: 28
zone3           -> junction5       -> elevation                      : 0.0                           ,
registers: [0, 0]              , address: 30

zone3           -> junction3       -> index                          : 37                            ,
registers: [16916, 0]          , address: 32
zone3           -> junction3       -> hydraulic_head                 : 514.2538452148438             ,
registers: [17408, 36927]      , address: 34
zone3           -> junction3       -> pressure                       : 514.2538452148438             ,
registers: [17408, 36927]      , address: 36
zone3           -> junction3       -> elevation                      : 0.0                           ,
registers: [0, 0]              , address: 38

zone3           -> junction4       -> index                          : 38                            ,
registers: [16920, 0]          , address: 40
zone3           -> junction4       -> hydraulic_head                 : 514.2538452148438             ,
registers: [17408, 36927]      , address: 42
zone3           -> junction4       -> pressure                       : 514.2538452148438             ,
registers: [17408, 36927]      , address: 44
zone3           -> junction4       -> elevation                      : 0.0                           ,
registers: [0, 0]              , address: 46

zone3           -> node40          -> index                          : 39                            ,
registers: [16924, 0]          , address: 48
zone3           -> node40          -> hydraulic_head                 : 514.2538452148438             ,
registers: [17408, 36927]      , address: 50
zone3           -> node40          -> pressure                       : 514.2538452148438             ,
registers: [17408, 36927]      , address: 52
zone3           -> node40          -> elevation                      : 0.0                           ,
registers: [0, 0]              , address: 54
[---TRUNCATED---]
```

- `def main():`

The `main` function ...

Function:

```python
def main():
    inp_file: str = parse_arguments()

    try:
        en: epanet = setup_epanet(inp_file)

        zones: set[str] = get_zones(en)

        clients: dict[str, ModbusTcpClient] = setup_clients(zones)

        en.openHydraulicAnalysis()
        en.initializeHydraulicAnalysis()

        while True:
            en.setTimeSimulationDuration(
                en.getTimeSimulationDuration() + en.getTimeHydraulicStep()
            )  # this way the duration is set to infinite.

            controls: dict = get_controls(clients, en)
            set_controls(en, controls)

            en.runHydraulicAnalysis()

            data: dict = read_data(en)
            write_data(clients, data)

            en.nextHydraulicAnalysisStep()

            time.sleep(1)
    except KeyboardInterrupt:
        print(">--- Program interrupted by user ---")
        sys.exit(0)  # clean exit confirmed by user action.
    except Exception as e:
        print(f"Failed to run EPANET simulation due to an unexpected error: {e}")
        sys.exit(1)
    finally:
        if "clients" in locals():
            for client in clients.values():
                client.close()
        if "en" in locals():
            en.closeHydraulicAnalysis()
            en.unload()
```

Something here...

Server receiving the registers like this:

Registers 0-1000 before without receiving anything:

```
holding registers (502): [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Registers 0-1000 after:

```
holding registers (502): [16576, 0, 50089, 37351, 50089, 37351, 0, 0, 16904, 0, 50089, 37351, 50089,
37351, 0, 0, 16908, 0, 50089, 37351, 50089, 37351, 0, 0, 16912, 0, 50089, 37351, 50089, 37351, 0, 0,
16916, 0, 50089, 37351, 50089, 37351, 0, 0, 16920, 0, 50089, 37351, 50089, 37351, 0, 0, 16924, 0, 50089,
37351, 50089, 37351, 0, 0, 16928, 0, 50089, 37351, 50089, 37351, 0, 0, 16932, 0, 50089, 37351, 50089,
37351, 0, 0, 16936, 0, 50089, 37351, 50089, 37351, 0, 0, 16940, 0, 50089, 37351, 50089, 37351, 0, 0,
16944, 0, 50089, 37351, 50089, 37351, 0, 0, 16948, 0, 50089, 37351, 50089, 37351, 0, 0, 16952, 0, 50089,
37351, 50089, 37351, 0, 0, 16956, 0, 50089, 37351, 50089, 37351, 0, 0, 16960, 0, 50089, 37351, 50089,
37351, 0, 0, 16964, 0, 50089, 37351, 50089, 37351, 0, 0, 16968, 0, 50089, 37351, 50089, 37351, 0, 0,
16972, 0, 50089, 37351, 50089, 37351, 0, 0, 16976, 0, 50089, 37351, 50089, 37351, 0, 0, 16980, 0, 50089,
37351, 50089, 37351, 0, 0, 16984, 0, 50089, 37351, 50089, 37351, 0, 0, 16988, 0, 50089, 37351, 50089,
37351, 0, 0, 16992, 0, 50089, 37351, 50089, 37351, 0, 0, 16996, 0, 50089, 37351, 50089, 37351, 0, 0,
17000, 0, 50089, 37351, 50089, 37351, 0, 0, 17004, 0, 50089, 37351, 50089, 37351, 0, 0, 17008, 0, 50089,
37351, 50089, 37351, 0, 0, 17012, 0, 50089, 37351, 50089, 37351, 0, 0, 17016, 0, 50089, 37351, 50089,
37351, 0, 0, 17020, 0, 50089, 37351, 50089, 37351, 0, 0, 17024, 0, 50089, 37351, 50089, 37351, 0, 0,
17026, 0, 50089, 37351, 50089, 37351, 0, 0, 17028, 0, 50089, 37351, 50089, 37351, 0, 0, 17030, 0, 50089,
37351, 50089, 37351, 0, 0, 17032, 0, 50089, 37351, 50089, 37351, 0, 0, 17034, 0, 50089, 37351, 50089,
37351, 0, 0, 17036, 0, 50089, 37351, 50089, 37351, 0, 0, 17038, 0, 50089, 37351, 50089, 37351, 0, 0,
17040, 0, 50089, 37351, 50089, 37351, 0, 0, 17042, 0, 50089, 37351, 50089, 37351, 0, 0, 17044, 0, 50089,
37351, 50089, 37351, 0, 0, 17046, 0, 50089, 37351, 50089, 37351, 0, 0, 17048, 0, 50089, 37351, 50089,
37351, 0, 0, 17050, 0, 50089, 37351, 50089, 37351, 0, 0, 17052, 0, 50089, 37351, 50089, 37351, 0, 0,
17054, 0, 50089, 37351, 50089, 37351, 0, 0, 17177, 0, 16968, 0, 16968, 0, 0, 0, 0, 0, 16968, 0, 16968, 0,
0, 0, 18367, 48995, 18367, 48995, 17165, 0, 0, 0, 0, 0, 16256, 0, 0, 0, 0, 0, 17168, 0, 0, 0, 0, 0,
16256, 0, 0, 0, 0, 0, 17171, 0, 0, 0, 0, 0, 16256, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```