

```

# pylint: disable=duplicate-code
from sunsynk import AMPS, CELSIUS, KWH, VOLT, WATT
from sunsynk.rwsensors import (
    NumberRWSensor,
    SelectRWSensor,
    SwitchRWSensor,
    SystemTimeRWSensor,
    TimeRWSensor,
)
from sunsynk.sensors import (
    BinarySensor,
    # EnumSensor,
    FaultSensor,
    InverterStateSensor,
    MathSensor,
    SDStatusSensor,
    Sensor,
    SensorDefinitions,
    SerialSensor,
    TempSensor,
)

PROG_CHARGE_OPTIONS = {
    0: "No Grid or Gen",
    1: "Allow Grid",
    2: "Allow Gen",
    3: "Allow Grid & Gen",
}

SENSORS = SensorDefinitions()

SENSORS += (
    BinarySensor(552, "Grid Connected", bitmask=1 << 2),

    Sensor(587, "587 Battery voltage", VOLT, 0.01),
    Sensor(607, "607 Total active power from side to side grid", WATT, -1),
    Sensor(608, "608 Grid side - inside total apparent power", WATT, -1),
    Sensor(619, "619 Out of Grid total power", WATT, -1),
    Sensor(620, "620 Out of Grid total apparent power", WATT, -1),
    Sensor(625, "625 Grid side total power", WATT, -1),
    Sensor(636, "636 Inverter output total Power", WATT, -1),
    MathSensor((607, 619), "Nonessential power new", WATT, factors=(-1, 1)),

    Sensor(598, "Grid L1 voltage", VOLT, 0.1), # aLineVolt
    Sensor(599, "Grid L2 voltage", VOLT, 0.1), # bLineVolt
    Sensor(600, "Grid L3 voltage", VOLT, 0.1), # cLineVolt
    Sensor(616, "Grid CT L1 power", WATT, -1), # aPower
    Sensor(617, "Grid CT L2 power", WATT, -1), # bPower
    Sensor(618, "Grid CT L3 power", WATT, -1), # cPower

    Sensor(644, "Load L1 voltage", VOLT, 0.1),
    Sensor(645, "Load L2 voltage", VOLT, 0.1),
    Sensor(646, "Load L3 voltage", VOLT, 0.1),

    Sensor(627, "Inverter L1 voltage", VOLT, 0.1),
    Sensor(628, "Inverter L2 voltage", VOLT, 0.1),
    Sensor(629, "Inverter L3 voltage", VOLT, 0.1),

    Sensor(630, "Inverter L1 current", AMPS, 0.01),
    Sensor(631, "Inverter L2 current", AMPS, 0.01),
    Sensor(632, "Inverter L3 current", AMPS, 0.01),

    Sensor(638, "Inverter frequency", "Hz", 0.01),

    # Sensor(643, "UPS power", WATT, 1),
    MathSensor((664, 665, 666), "GEN power", WATT, factors=(1, 1, 1)),
    # Sensor(166, "AUX power", WATT, 1),

    SelectRWSensor(172, "Prog1 charge", options=PROG_CHARGE_OPTIONS),

```

```
SelectRWSensor(173, "Prog2 charge", options=PROG_CHARGE_OPTIONS),
SelectRWSensor(174, "Prog3 charge", options=PROG_CHARGE_OPTIONS),
SelectRWSensor(175, "Prog4 charge", options=PROG_CHARGE_OPTIONS),
SelectRWSensor(176, "Prog5 charge", options=PROG_CHARGE_OPTIONS),
SelectRWSensor(177, "Prog6 charge", options=PROG_CHARGE_OPTIONS),

Sensor(643, "Essential power", WATT, 1),
MathSensor((653, 643), "Nonessential power", WATT, factors=(1, -1)), # Grid load power -
Grid ups power
MathSensor((672, 673, 674, 675), "PV power", WATT, factors=(1, 1, 1, 1)),

#   Sensor(2, "Comm protocol version"),

)
```