

Lecture 16 - Using Objects

1015 Lect 16 pt 1 - More on Objects - <https://youtu.be/gfla1aTjTis>

1015 Lect 16 pt 2 - More on Objects - https://youtu.be/_zklj-dHi0A

From Amazon S3 - for download (same as youtube videos)

1015 Lect 16 pt 1 - More on Objects

1015 Lect 16 pt 2 - More on Objects

This is always a complex thing - you look at objects and say, "why would I ever use this?". This is a valid consideration. Objects are good when you have a more complex system with lots of moving parts. An introductory class makes that difficulty. Last year I did green house controllers - because I am interested in them. This year we are going to talk about solar, batteries, vans and other fun stuff.

First batteries. This is where you get to find out about one of the "fun" things about being in Compeer Science. Writing pregames means that you get to learn about non-computer science subjects. To build an effective chunk of software that works with batteries - you have to learn about batteries. Most people look at batteries and kind of have the view that you just hook them up and they work. Not really. They are complicated!

First of all - the battery in your car - the starter battery. It is designed to produce a large output for a short time and then get re-charged. When you store electricity from a solar system you want to produce a small out put for a large time. So car batteries are the wrong thing for solar storage.

They make "led-acid" batteries for small output over large time. This are classified as "golf-cart" batteries.

The second thing is how batteries are rated. Let's say you have a 100 Ampere Hour battery. This means that from 100% to 0% on the battery you get 100 Ampere Hours (AH). The voltage will go from 14.6 volts to 11.8 volts on a led-acid battery. This voltage is determined by the chemistry of he batter. If you actually use your batteries in this fashion, from 100% down to 0% you will get about 5 charge / discharge cycles and then need to replace the battery. They are expensive - so you don't want to do that. You want to use the battery in a way where you get a high number of charge/discharge cycles. For a led-acid battery this is from 95% charge to 75% charge. So your 100 AH battery actually can only be used in this 20% range and $20\% * 100 \text{ AH} = 20 \text{ AH}$ useful power. Ouch!

To convert from Ah to Watts - you take voltage times amperage to get watts.

$$W = A * v$$

Watts is a measure of how much useful work you are getting. Now if you had a battery where you had a bigger useful range - and more charge/discharge cycles that would be good. There is such a thing. That thing is a Lithium Ion battery. It would be nice if there was just one Lithium Ion type of battery but there are actually a big bunch of them. They vary in voltage, charge cycles, weight, mechanical properties - and what happens if exposed to air or water. Some of them will explode on contact with air - some will catch on fire and burn when exposed to water - some will only slowly smolder if punctured. Some expand when they burn. Some let off toxic gasses.

If you are going to use one in a vehicle - then you need to take into consideration that in an accident the battery may get damaged. If you use a chemical combination that catches on fire - this turns an accident into a potentially deadly fire! Lithium Phosphate LiPo is the lightest and highest energy - but if you puncture the battery there is no way to put it out. It will burn water! It will burn air - it will burn steel! Very bad! This is the kind of a battery in your laptop and your phone!

Lithium Iron Phosphate (LiFePO_4) is a little bit heavier - but it will only smolder. It will not burst into open flame and it can be put out by dousing with water or foam or other flame retardants.

But batteries have a temperature range where they work. Lead-acid is -30 degrees Fahrenheit to 200 degrees F. That is a lot. No such luck with LiFePO_4 . Most of them will be damaged permanently if you try to charge them below 40 degrees and they will be messed up in weird ways if you try to charge at over 120 degrees.

So if you want to use the LiFePO_4 in a "van" you will need to heat and cool the batteries when they are being charged.

Now to cool the battery you on medium days you can just blow a fan over it. Take cool air and use that.

On hot days you need to air condition the battery. Basically pump heat from it to some other location.

On cold days you need to heat the battery up. So you need to turn on a heater and warm it up.

Fortunately this Van has an engine block heater that runs off of diesel and can be started. Instead of heating the engine block - the hot fluids can be sent to where the batteries are to heat the batteries.

The engine air conditioner is not so simple. This would require the use of the main engine to run it. So we need a dedicated air conditioner that can run and pump cold air over the batteries.

So an outside air conditioner is the usual method for cooling the battery. Since it is small AC the best way to do this is to put just the battery in an insulated box and just provide AC to just the battery when the outside air temperature exceeds the desired charge temperature of the battery.

When the battery is too warm or too cold then we turn off the charge and protect the battery. This means that we need a switch to shut off the charging.

When LiFePo4 batteries discharge they produce heat also. Again we will need to cool them. So as a part of the discharge process we will run the AC to cool the battery.

Keeping proper charging and temperature gives us a battery that we can use between 85% and 15%, or 70% of its ampere hour capacity with between 4,000 and 10,000 cycles.

So you can get 100 to 200 cycles from a \$200 battery, and 4,000 to 10,000 cycles out of a \$1000 battery and a slew of outside systems to monitor and maintain the system. Over the life cycle of the system the LiFePo4 battery much cheaper, but only if you can automate the "system".

So what is in our system:

1. A AC to cool the battery - this can be turned on/off by the computer
2. A way to turn on the "heater" - this can be turned on/off by the computer
3. A temperature sensor for the battery
4. A temperature sensor for the outside air
5. A sensor that says when the "charger" is providing electricity to charge the battery
6. A sensor that says when the user is pulling power from the battery
7. A switch - computer controlled - to turn off the charger
8. A switch - computer controlled - to turn off user - in case the battery is to empty
9. The "fan" in the battery compartment
10. The "battery"
11. A battery charge sensor (how full is the battery)

For objects we have:

`class switch` - a controller that turns on/off a switch

`class charger_switch (switch)` - an inherited switch that can turn off the charging

`class output_switch (switch)` - an inherited switch that can turn off usage of the battery if power is too low.

`class ac_control` - a class to turn on/off the AC system

`class temperature_sensor` - a class to measure the temperature

`class battery_temperature (temperature_sensor)` - the temperature of the battery

`class outside_temperature (temperature_sensor)` - the temperature of the outside air

class battery_state - how full is the battery

class charger_state - is the charger available

class fan - a class to turn on/off the fan

class user_poser - a class that shows how much power the user is using

So let's start looking at individual classes.

```
import simulated_system

class switch:

    def __init__ ( self, addr, name ):
        self.my_addr = addr
        self.name = name
        self.state = 0                # Assume that the switch is off
        simulated_system.system_set ( f"switch:{self.my_addr}", 0 )    # Set to off

    def isOn ( self ):
        if self.state == 0:
            return False
        else:
            return True

    def turnOn ( self ):
        self.state = 1
        simulated_system.system_set ( f"switch:{self.my_addr}", 1 )    # Set to on

    def turnOff ( self ):
        self.state = 0
        simulated_system.system_set ( f"switch:{self.my_addr}", 0 )    # Set to off

    def __str__ ( self ):
        if self.isOn():
            return f"Switch {self.name} at {self.my_addr} is ON"
        return f"Switch {self.name} at {self.my_addr} is off"

    def myName( self ) :
        return self.name

if __name__ == "__main__":
    print ( "In Main" )

    s = switch(4)
    print ( f"{s}" )
```

Now let's take a look at the 2 different switches:

```
import switch

class charger_switch ( switch.switch ):

    def __init__ ( self ):
        self.my_addr = 1
        super().__init__ ( 1, "charger" )

if __name__ == "__main__":
    print ( "In Main" )

    s = charger_switch()
    print ( f"{s}" )
```

Now let's take a look at the other one:

```
import switch

class output_switch ( switch.switch ):

    def __init__ ( self ):
        self.my_addr = 2
        super().__init__ ( 2, "output" )

if __name__ == "__main__":
    print ( "In Main" )

    s = output_switch()
    print ( f"{s}" )
```

Takeaways

1. Batteries for solar storage is not a "simple" technology. You don't just hook up some solar to some batteries and it works.
2. Details are really important.
3. We didn't talk about the complexity of "solar" power - or how you want to control the AC or the "heater".
4. A class "hierarchy" will require a "significant" knowledge of the underlying system you are going to control.
5. The "class" or Object Oriented Programming (OOP) provides a useful way to "model" the system - and to re-use code (like the temperature sensor)
6. If you want to use "solar" and "batteries" to build an "off" grid house or van - you need to have a "lot" of complex technology and knowledge about how the system works.