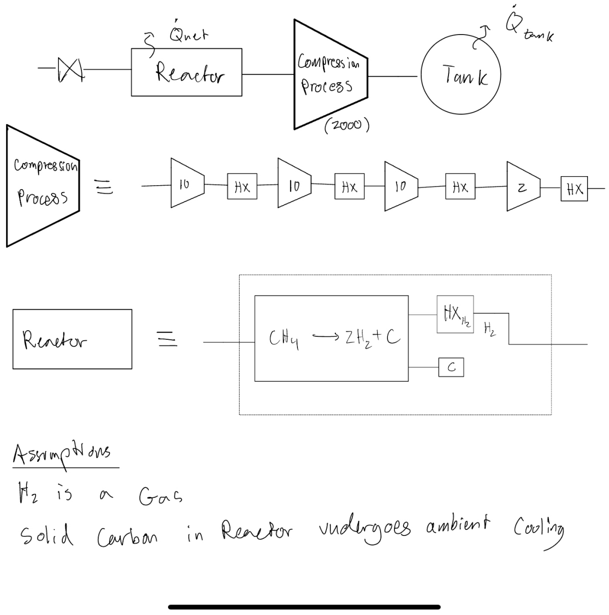
```julia
begin
    using Unitful ✓
    using CoolProp ✓
end
```

Project setup

Diagram 1

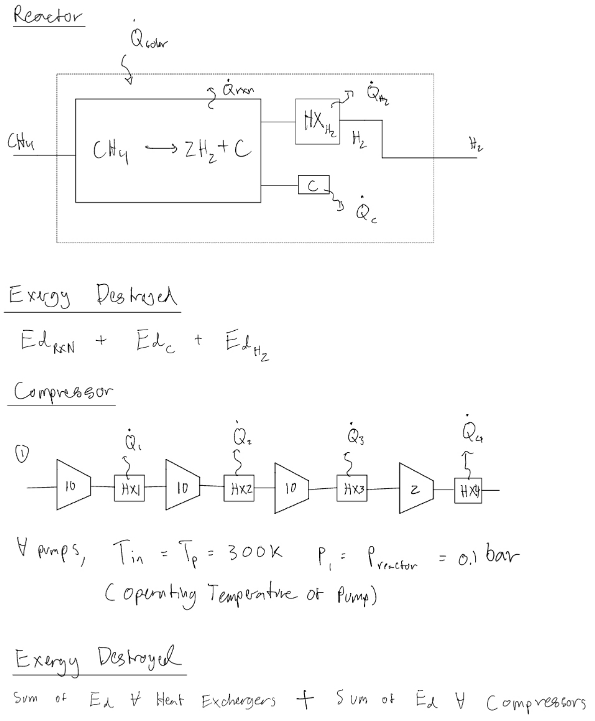


Assumptions

H_2 is a Gas

Solid Carbon in Reactor undergoes ambient Cooling

Diagram 2



Exergy Destroyed

$Ed_{RXN} + Ed_C + Ed_{H_2}$

$\forall$ pumps, $T_{in} = T_p = 300K$ $P_1 = P_{reactor} = 0.1$ bar

(Operating Temperature of Pump)

Exergy Destroyed

Sum of Ed $\forall$ Heat Exchergers + Sum of Ed $\forall$ Compressors

```julia
# Adiabatic H2 Compressor
struct Pump
    W::Any
    ratio::AbstractFloat
    P::Any
    T::Any
    name::AbstractString
    η::Number

end
```

```julia
# Isenthalpic CH4 Valve
struct Valve
    Tin::Any
    Pin::Any
    Pex::Any
    name::AbstractString
end
```

```
# ideal Heat Exhanger, approximatly isobaric cooling
struct Cooler
    Tin::Any
    Tex::Any
    P::Any
    Tb::Any
    name::AbstractString
end
```

```
# CH4 decomposition reactor
# Reactor does no work
struct Reactor
    Tin::Any
    Tex::Any
    P::Any
    Tb::Any
end
```

```
0.01377946 mol s⁻¹
```

```
# Assumptions
begin

    # Reactions is complete: CH4(g) --> 2H2(g) + C(s)
    # Isenthalpic throttling valve

    # Given Constants
    P_in = 1u"atm"
    T_in = 298u"K"
    P_react = 10u"kPa"
    T_tank = 300u"K"
    P_tank_min = 200u"bar"
    P_tank_max = 220u"bar"
    V_tank = 1u"m^3"
    T_pyro = 1600u"K"

    Q_solar = 15u"kW"
    H_rxn = 74.6u"kJ/mol"
    Hf_meth = -74850u"kJ/kmol"
    M_h2 = 2*1.00794u"g/mol"
    M_meth = 16.04u"g/mol"
    M_carbon = 12u"g/mol"
    C_carbon = 0.71u"J/g/K"
    R = 8.3145u"J/mol/K"

    MolarMass = Dict("C"=>12.01u"g/mol", "H2"=>2.02u"g/mol", "CH4"=>16.04u"g/mol")
    MODEL2 = Dict("C"=>410_260u"kJ/kmol", "H2"=>236_100u"kJ/kmol",
    "CH4"=>831_650u"kJ/kmol")

    ## Initial Calcuations
    n_dot_meth = uconvert(u"kmol/s", 0.1 * Q_solar / H_rxn)
    n_dot_meth |> println
    n_dot_h2 = 2 * n_dot_meth
    n_dot_c = n_dot_meth
    n_dot_meth = 0.00688973u"mol/s"
    n_dot_h2 = 2 * n_dot_meth

end

```

```
2.0107238605898127e-5 kmol s⁻¹
```

```
0.006938662225922842 mol s⁻¹
```

```
begin
    massH2 = 1u"kg"
    massC = 3u"kg"

    day = 10u"hr"
    molH2 = massH2/MolarMass["H2"]
    molC = massC/MolarMass["C"]
    molCH4 = molC

    MolarFlow = Dict("C"=>molC/day, "H2"=>molH2/day, "CH4"=>molC/day)
    MolarFlow["C"] = uconvert(u"mol/s", MolarFlow["C"])
    MolarFlow["H2"] = uconvert(u"mol/s", MolarFlow["H2"])
    MolarFlow["CH4"] = uconvert(u"mol/s", MolarFlow["CH4"])

end
```

1. Conduct a search of available hydrogen compressors. Does a single compressor exist that serves the required function (gas flow, pressure ratio)? If not, what multiple compressor configuration would satisfy the specifications? Explain and include links where appropriate.

```
md"""
1) Conduct a search of available hydrogen compressors. Does a single compressor exist
that serves the required function (gas flow, pressure ratio)? If not, what multiple
compressor configuration would satisfy the specifications? Explain and include links
where appropriate.
"""
```

```
let
    "E = 0.3" |> println
    info(0.3)
    "E = 0.9" |> println
    info(0.9)

end
```

```
E = 0.3
Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 10.0, 10 kPa, 35 °C,
 "H2", 0.3)
temperature(pump) = 1245.0058252297158 K
exergyDestroyed(pump, Tₒ, Pₒ) = 6606.741313859118 kJ kmol⁻¹

Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 10.0, 100.0 kPa, 35 °
C, "H2", 0.3)
temperature(pump) = 1245.968599004329 K
exergyDestroyed(pump, Tₒ, Pₒ) = 6614.87804607824 kJ kmol⁻¹

Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 10.0, 1000.0 kPa, 35 °
C, "H2", 0.3)
temperature(pump) = 1255.6406617726086 K
exergyDestroyed(pump, Tₒ, Pₒ) = 6696.212299640985 kJ kmol⁻¹

Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 2.0, 10000.0 kPa, 35 °
C, "H2", 0.3)
temperature(pump) = 542.7954111093296 K
exergyDestroyed(pump, Tₒ, Pₒ) = 3241.554839046439 kJ kmol⁻¹

E = 0.9
Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 10.0, 10 kPa, 35 °C,
 "H2", 0.9)
temperature(pump) = 626.6619808294671 K
exergyDestroyed(pump, Tₒ, Pₒ) = 469.87436105212873 kJ kmol⁻¹

Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 10.0, 100.0 kPa, 35 °
C, "H2", 0.9)
temperature(pump) = 626.7046199045352 K
exergyDestroyed(pump, Tₒ, Pₒ) = 470.85399218199564 kJ kmol⁻¹

Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 10.0, 1000.0 kPa, 35 °
C, "H2", 0.9)
temperature(pump) = 627.1372194849134 K
exergyDestroyed(pump, Tₒ, Pₒ) = 480.94793904540893 kJ kmol⁻¹

Main.workspace#3.Pump(-362.85892190259995 s kW mol⁻¹, 2.0, 10000.0 kPa, 35 °
C, "H2", 0.9)
temperature(pump) = 383.5751800075623 K
exergyDestroyed(pump, Tₒ, Pₒ) = 182.84843939503662 kJ kmol⁻¹
```

2. What size of methane mass flow controller (in sccm) is required at nominal efficiency? Find a suitable product.

```
ans2 = 26956.939498993706 cm³ minute⁻¹
```

```
ans2 = let
    molarFlowCH4 = uconvert(u"kmol/s", 0.1 * Q_solar / H_rxn)
    ρ = PropsSI("D", "T", 273u"K", "P", 1u"atm", "CH4")
    scfm = molarFlowCH4*MolarMass["CH4"] / ρ
    uconvert(u"cm^3/minute", scfm)
end
```

3. For your compressor configuration in part A, estimate the overall isentropic efficiency of the compression process.

We couldn't find any good data for our specific hydrogen pump, but various papers suggest an isentropic effeciency of 65-85%.

```
# ans3 = let
#   pump_size = -1u"kW"
#   pump = Pump(pump_size/MolarFlow["H2"], 10, P_react, 35u"°C", "H2", 0.8)
#   @show temperature(pump)
# end
```

4. Calculate the final pressure of the tube cylinder at the end of one day.

```
ans4 = 212.37350437526044 bar
```

```
ans4 = let
    molH2 = 1u"kg"/M_h2
    P_H2 = molH2 * R * T_tank / V_tank
    P_final = uconvert(u"bar", P_tank_min + P_H2)
end
```

5. Calculate the rate of exergy destruction at the mass flow controller at nominal efficiency

```
ans5 = 0.039785264651772234 kW
```

```
ans5 = let
    Tₒ = 298u"K"
    Pₒ = 1u"atm"
    valve = Valve(T_in, P_in, P_react, "CH4")
    ED = exergyDestroyed(valve, Tₒ, Pₒ)*MolarFlow["CH4"]
    uconvert(u"kW", ED)
end
```

6. Calculate the rate of exergy destruction in the reactor at nominal efficiency

```
ED_reactor = 0.2148653074468528 kW
```

```
ED_reactor = let
    Tₒ = 298u"K"
    Pₒ = 1u"atm"
    valve = Valve(T_in, P_in, P_react, "CH4")
    T_in_to_react = temperature(valve)

    Q_in = MolarFlow["CH4"]*H_rxn/0.1

    Tb = T_pyro
    react = Reactor(T_in_to_react, T_pyro, P_react, Tb)

    chem_Ex = MODEL2["CH4"] - 2*MODEL2["H2"] - MODEL2["C"]
    e_in = get_flow_exergy(T_in_to_react, P_react, Tₒ, Pₒ, "CH4")
    e_out = get_flow_exergy(T_pyro, P_react, Tₒ, Pₒ, "H2")
    flow_exergy = uconvert(u"kW", e_in*MolarFlow["CH4"] - e_out*MolarFlow["H2"])

    flow_exergy

    Q_rxn = heat(react)
    Q_waste = uconvert(u"kW", Q_in - Q_rxn)
    Q_net = Q_rxn

    Ed = (1-Tₒ/Tb)*Q_net + flow_exergy + chem_Ex*MolarFlow["CH4"]
    uconvert(u"kW", Ed)

end
```

7. What fraction of this reactor exergy destruction is consumed by cooling the graphite from 1600 K to room temperature?

```
ED_C = 0.0005135666666666699 kW
```

```
ED_C = let
    Tₒ = 298u"K"
    Pₒ = 1u"atm"
    Tb = 300u"K"
    massCarbon = 3u"kg"
    ΔT = (T_pyro - Tₒ)
    Q = massCarbon*C_carbon*ΔT/10u"hr"
    uconvert(u"kW", (1-Tₒ/Tb)*Q)


end
```

```
0.239017956304417
```

```
(ED_C/ED_reactor)*100
```

Answer:

0.23% of exergy destruction consumed by cooling graphite from 1600K to room temperature.

8. What fraction of this reactor exergy destruction is consumed by cooling the product hydrogen from 1600 K to room temperature?

```
ED_H2 = 337.8492550952602 kg m² s⁻³
```

```
ED_H2 = let
    Tₒ = 298u"K"
    Pₒ = 1u"atm"
    Tb = Tₒ

    H1 = PropsSI("Hmolar", "T", T_pyro, "P", P_react, "H2")
    H2 = PropsSI("Hmolar", "T", Tₒ, "P", P_react, "H2")
    Q = (H2-H1)*MolarFlow["H2"]
    e1 = get_flow_exergy(T_pyro, P_react, Tₒ, Pₒ, "H2")
    e2 = get_flow_exergy(Tₒ, P_react, Tₒ, Pₒ, "H2")

    (1-Tₒ/Tb)*Q + (e1-e2)*MolarFlow["H2"]
end
```

```
ans8 = 1.5723769421400318
```

```
ans8 = let
    uconvert(u"K/K", ED_H2/ED_reactor)
end
```

Answer

158% of the exergy destruction of the reactor is destroyed cooling H2.

9. Calculate the rate of exergy destruction in the compression process.

```
ans9 = 0.5262926583442024 kW
```

```
ans9 = let
    Tₒ = 298u"K"
    Pₒ = 1u"atm"
    Tb = 300u"K"
    Tp = 300u"K"
    P = P_react
    T = Tp
    power = -5u"kW"
    pumps = []
    hxs = []
    P = P_react
    T = Tp
    for i = 1:3
        pump = Pump(power/MolarFlow["H2"], 10, P, Tp, "H2", 0.65)
        push!(pumps, pump)
        P = pump.P * pump.ratio
        T = temperature(pump)
        push!(hxs, Cooler(T, Tp, P, Tb, "H2"))
    end
    finalPump = Pump(power/MolarFlow["H2"], 2, P, Tp, "H2", 0.65)
    push!(pumps, finalPump)
    T = temperature(finalPump)
    push!(hxs, Cooler(T, Tp, finalPump.ratio*P, Tb, "H2"))

    Ed_pumps = sum([exergyDestroyed(pump, Tₒ, Pₒ) for pump in pumps])
    Ed_hxs = sum([exergyDestroyed(hx, Tₒ, Pₒ) for hx in hxs])
    Ed_compression = MolarFlow["H2"] * (Ed_pumps + Ed_hxs)
    uconvert(u"kW", Ed_compression)
end
```

10. Calculate the total exergy destroyed as the pressurized hydrogen loses heat from the tank to the surroundings

Our design uses heat exchangers before and after every compressor stage. This means that the gas enters the tank at ambient temperature and tank pressure. Therefore there is not exergy change/no exergy destruction at this point.

Answer: 0

11:

If the heat lost by the hydrogen stream between the reaction zone and the compressor inlet could be sent completely back into the reaction zone, then by what percentage could the methane mass flow rate increase? What would be the exergy destruction? Would it increase or decrease? By how much?

```
10.474841264052992
```

```
let
    h1 = PropsSI("Hmolar", "T", T_pyro, "P", P_react, "H2")
    h2 = PropsSI("Hmolar", "T", 298u"K", "P", 200u"bar", "H2")

    Q_cooling = (h1-h2)*n_dot_h2

    Q_inital = uconvert(u"kW", MolarFlow["CH4"]*H_rxn/0.1)
    Q_final = uconvert(u"kW", Q_inital + Q_cooling)

    n_final_new = uconvert(u"mol/s", Q_final*0.1/H_rxn)
    uconvert(u"K/K", (n_final_new-MolarFlow["CH4"])/MolarFlow["CH4"])*100
end
```

Answer: The molar (and mass) flow rate of methane would increase by 10.5%.

12.

What are the technical and/or societal benefits and drawbacks of using solar heating in this design, as compared to burning methane or hydrogen for pyrolysis?

```
md"""
### 12.
What are the technical and/or societal benefits and drawbacks of using solar heating
in this design, as compared to burning methane or hydrogen for pyrolysis?
"""
```

It is assumed that a solar concentrator is being used for this system. A solar concentrator works by having a curved surface or a series of flat surfaces all angled toward one focus, so that the sunlight bounces of the surfaces and reflects all to the same point. At that point a receiver is placed to collect the energy from the sun. This set up is fairly cheap to manufacture and set up. However, there is no way to store the energy so the solar concentrator can only be used when the sun is out and shining. It would be ideal to position this somewhere that the efficiency of this process is not decreased significantly.

If solar panels were being used, a battery can be attached to store the energy that isn't being used during the day, allowing the process to continue, even when the sun it not present. While having a battery solves the issue of the sun's limited availablility as an energy source, a new issue is created, cost. The website for the Office of Energy Efficiency and Renewable Energy states that solar batteries can range from $12,000 to 22,000. In general, there is a high upfront cost with solar panels due to the price of purchasing the panels and their installation if that is required. However, this cost could be made up through the money saved when not paying for other energy sources. The US Government is also starting to offer incentives for using solar panels so those may aid in the initial cost of the solar energy set up.

If the hydrogen and graphite being produced for the assumed 10 hours a day is suffcient, and an irregular production rate due to the sun exposure fluctuation is no issue, than a solar concentrator is a great option. The most significant benefit of solar power over other energy sources, such as burning methane, is that the sun is a renewable energy source. It will never run out of heat to provide, at least not for roughly 5 billion more years. It is also a green energy source. The use of solar energy decreases greenhouse gas production which is vital to slowing the effects of climate change.

Sources:

https://lightningsolar.com.au/comparative-guide-advantages-disadvantages-of-solar-panels/

https://powersolarphoenix.com/commercial-solar-panels-cost/

https://www.energy.gov/eere/solar/articles/should-i-get-battery-storage-my-solar-energy-system

13.

More broadly, what are the technical and societal benefits and drawbacks of using methane (from natural gas) as a source for graphite and hydrogen?

There is no shortage of methane production in our society. The farming and agriculture industries are huge contributors to the methane content in the atmosphere. When manure is produced by livestock, such as pigs or cattle, methane is released. When the manure is spread onto crops for fertilization, methane is released. A fight has begun to reduce meat and dairy consumption in the US has begun. Alternative milks and vegetarian/vegan options are becoming more widespread than ever before. It is apparent however that much more time and effort is needed to change the general population's eating habits so methane production continues.

Another significant source of methane in our society is landfills. As organic materials decompose under piles of waste, there are not significant sources of oxygen to facilitate aerobic decomposition. Bacteria that perform anaerobic decomposition come into play. These bacteria are huge sources of methane through their decomposition processes. Composting can help decrease these methane emissions. By collecting organic materials and allowing them to decompose in environments that provide aeration, organisms that do not produce methane can process the materials. Composting is not a widespread habit. Many communities do not have access to composting programs and especially programs that accept food waste, so much of the organic waste still ends up in landfills, producing methane.

Methane is a significant cause for the trapping of heat within Earth's atmosphere that is leading to climate change. According to UNECE, methane has a warming potential roughly 28-34% higher than that of CO2. Methane capture is a solution being presented to the high emissions of methane from the above sources. In that set up, perforated tubes are arranged through the methane producing areas to collect the methane and transport it to facilities that will burn it. This will prevent much of the methane being produced in industrial settings from reaching the atmosphere.

A downside of burning methane is that CO2 is produced by the process. CO2 in the atmosphere is another cause to the greenhouse effect. However, if the methane being burned is harvested from the atmosphere then the benefits of destroying methane outweighs the drawbacks of CO2 production. The burning of this collected methane is also beneficial because it eliminates reliance on other energy sources that also produce greenhouse gasses. As outlined above, due to the processes used for food production and waste management that this society relies upon, there is an abundance of methane to be utilized for burning and heating. If it is not utilized, it is only making the climate change issue worse.

Sources:

https://www.epa.gov/lmop/basic-information-about-landfill-gas

https://unece.org/challenge

```
temperature (generic function with 2 methods)
```

```julia
begin
    function temperature(pump::Pump)
        k = 1.4
        polytropicExp = (k-1)/k
        Ts = uconvert(u"K", pump.T)*(pump.ratio)^polytropicExp
        Hs = PropsSI("Hmolar", "T", Ts, "P", pump.P*pump.ratio, pump.name)
        H = PropsSI("Hmolar", "T", pump.T, "P", pump.P, pump.name)
        Hr = H + (Hs-H)/pump.η
        P2 = pump.P*pump.ratio
        T = PropsSI("T", "P", P2, "Hmolar", Hr, pump.name)
        return uconvert(u"K", T)
    end

    function temperature(valve::Valve)
        Hin = PropsSI("H", "T", valve.Tin, "P", valve.Pin, valve.name)
        return uconvert(u"K", PropsSI("T", "H", Hin, "P", valve.Pex, valve.name))
    end
end
```

```
pressure (generic function with 1 method)
```

```julia
begin
    function pressure(hx::Cooler)
        k = 1.4
        polytropicExp = k/(k-1)
        P = hx.P*(hx.Tex/hx.Tin)^polytropicExp
        return uconvert(u"bar", P)
    end
end
```

```
exergyDestroyed (generic function with 4 methods)
```

```julia
begin
    function exergyDestroyed(pump::Pump, Tₒ, Pₒ)
        T = temperature(pump)
        P = pump.ratio*pump.P
        e_in = get_flow_exergy(pump.T, pump.P, Tₒ, Pₒ, pump.name)
        e_ex = get_flow_exergy(T, P, Tₒ, Pₒ, pump.name)

        Hex = PropsSI("Hmolar", "T", T, "P", P, pump.name)
        Hin = PropsSI("Hmolar", "T", pump.T, "P", pump.P, pump.name)
        return uconvert(u"kJ/kmol", Hex-Hin + (e_in - e_ex))
    end

    function exergyDestroyed(valve::Valve, Tₒ, Pₒ)
        Tex = temperature(valve)
        e_in = get_flow_exergy(valve.Tin, valve.Pin, Tₒ, Pₒ, valve.name)
        e_ex = get_flow_exergy(Tex, valve.Pex, Tₒ, Pₒ, valve.name)
        return uconvert(u"kJ/mol", (e_in - e_ex))
    end

    function exergyDestroyed(react::Reactor, Tₒ, Pₒ)
        Tb = (react.Tex + react.Tin)/2
        e_in = get_flow_exergy(react.Tin,react.P,Tₒ,Pₒ,"CH4")
        e_ex = get_flow_exergy(react.Tex,react.P,Tₒ,Pₒ,"H2")
        chem_Ex = MODEL2["CH4"] - 2*MODEL2["H2"] - MODEL2["C"]
        Q = -heat(react)
        Ed = (1-Tₒ/Tb)*Q + MolarFlow["CH4"]*e_in-MolarFlow["H2"]*e_ex +
        MolarFlow["CH4"]*chem_Ex
        return uconvert(u"kW", Ed)
    end

    function exergyDestroyed(hx::Cooler, Tₒ, Pₒ)
        e_in = get_flow_exergy(hx.Tin, hx.P, Tₒ, Pₒ, hx.name)
        e_ex = get_flow_exergy(hx.Tex, hx.P, Tₒ, Pₒ, hx.name)
        Tb = (hx.Tin + hx.Tex)/2
        Q = -heat(hx)
        Ex = (1-Tₒ/Tb)*Q + e_in - e_ex
        return uconvert(u"kJ/mol", Ex)
    end
end
```

```
heat (generic function with 2 methods)
```

```julia
begin
    function heat(react::Reactor)
        Hin = molar_enthalpy(react.Tin, react.P, "CH4", Hf_meth)
        Hex = molar_enthalpy(react.Tex, react.P, "H2")
        Q =  n_dot_h2*Hex-MolarFlow["CH4"]*Hin
        return uconvert(u"kW", Q)
    end

    function heat(hx::Cooler)
        Hin = PropsSI("Hmolar", "T", hx.Tin, "P", hx.P, hx.name)
        Hex = PropsSI("Hmolar", "T", hx.Tex, "P", hx.P, hx.name)
        return uconvert(u"kJ/mol", (Hex-Hin))
    end
end
```

```julia
let
    #Tₒ = 298u"K"
    #hx = Cooler(T_pyro, Tₒ, P_react, T_pyro, "H2")
    #uconvert(u"bar/bar", P_react/pressure(hx))
end
```

```
molar_enthalpy (generic function with 2 methods)
```

```julia
function molar_enthalpy(T, P, name::String, Hf=0u"kJ/kmol")
    Href = PropsSI("Hmolar", "T", 298.15u"K", "P", 1u"atm", name)
    H =  PropsSI("Hmolar", "T", T, "P", P, name)
    return uconvert(u"kJ/kmol", Hf + H - Href)
end
```

```
get_flow_rate (generic function with 3 methods)
```

```julia
function get_flow_rate(m_dot, T=273.15u"K", P=1u"atm")
    ρ = PropsSI("D", "T", T, "P", P, "CH4")
    return uconvert(u"ft^3/minute", m_dot/ρ)
end
```

```
get_flow_exergy (generic function with 1 method)
```

```julia
function get_flow_exergy(T, P, Tₒ, Pₒ, name::String)
    Hₒ = PropsSI("Hmolar", "T", Tₒ, "P", Pₒ, name)
    Sₒ = PropsSI("Smolar", "T", Tₒ, "P", Pₒ, name)
    H = PropsSI("Hmolar", "T", T, "P", P, name)
    S = PropsSI("Smolar", "T", T, "P", P, name)
    flow = H - Hₒ - Tₒ*(S - Sₒ)
    return uconvert(u"kJ/kmol", +flow)
end
```

```
info (generic function with 1 method)
```

```julia
function info(η)
        Tₒ = 298u"K"
        Pₒ = 1u"atm"

        power = -5u"kW"
        pumps = []
        P = P_react
        for i = 1:3
            pump = Pump(power/n_dot_h2, 10, P, 35u"°C", "H2", η)
            push!(pumps, pump)
            P = pump.P * pump.ratio
        end
        push!(pumps, Pump(power/n_dot_h2, 2, P, 35u"°C", "H2", η))

        for pump in pumps
            pump |> println
            @show temperature(pump)
            @show exergyDestroyed(pump, Tₒ, Pₒ)
            "\n" |> print
        end
end
```

Resources: [here] https://www.hydrogen.energy.gov/pdfs/9013*energy*requirements*for*hydrogen*gas*compression.pdf

```julia
md"""

Resources: [here] https://www.hydrogen.energy.gov/pdfs/9013_energy_requirements_for_hydrogen_gas_compression.pdf

"""
```