

Contents

- Fermeture des figures
- Déclarations
- BMP180
- HCSR04
- TD1208R
- STM32
- Batterie
- Chargeur
- Cellule solaire
- Courbe solaire
- Duree sur batterie
- Affichages

Fermeture des figures

```
clc
clear all
close all
```

Déclarations

```
t_max = 5*24; % Temps max en heures
t = 0:1:60*t_max; % Echelle de temps en heures

E_batt_t = zeros(length(t),1); % Initialisation de vecteur
ensol = zeros(length(t),1); % Initialisation de vecteur

Sunrise = 9; % Heure de lever du soleil
Sunset = 17; % Heure de coucher du soleil
T_soleil = Sunset - Sunrise; % Heures de soleil par jour
Midday = (Sunrise+(Sunset-Sunrise)/2); % Heure du zenith

DeepSleep = 20; % DeepSleep en secondes
Time_on = 350e-3; % Temps reveillé en secondes
% mesuré grace à des timers

Cycle_complet = DeepSleep + Time_on; % Durée d'un cycle complet

Rapport_on = Time_on / Cycle_complet; % Rapport système reveillé
Rapport_off = DeepSleep / Cycle_complet; % Rapport système en DeepSleep
```

BMP180

```
% Calcul de la consommation du capteur de température et pression. En se
% basant sur la datasheet. P_BMP_moy représente la puissance moyenne
% consommée par ce capteur en fonction de la durée où il est allumé et la
% durée où il est en DeepSleep.

I_BMP_on = 5e-6; % Amperes
```

```

I_BMP_off = 3e-6; % Amperes
U_BMP = 3; % Volt

P_BMP_on = U_BMP * I_BMP_on; % Watt
P_BMP_off = U_BMP * I_BMP_off; % Watt

P_BMP_moy = P_BMP_on * Rapport_on + P_BMP_off * Rapport_off; % Watt

```

HCSR04

```

% Calcul de la consommation du capteur Ultrason. En se basant sur la
% datasheet. P_HCSR04_moy représente la puissance moyenne consommée par ce
% capteur en fonction de la durée où il est allumé et la durée où il est en
% DeepSleep.

I_HCSR04_on = 15e-3; % Ampères
I_HCSR04_off = 2e-3; % Ampères
U_HCSR04 = 5; % Volt

P_HCSR04_on = U_HCSR04 * I_HCSR04_on; % Watt
P_HCSR04_off = U_HCSR04 * I_HCSR04_off; % Watt

P_HCSR04_moy = P_HCSR04_on * Rapport_on + ...
               P_HCSR04_off * Rapport_off; % Watt

```

TD1208R

```

% Calcul de la consommation du module Sigfox. En se basant sur la
% datasheet. P_TD1208R_moy représente la puissance moyenne consommée par ce
% module en fonction de la durée où il est allumé et la durée où il est en
% DeepSleep.

I_TD1208R_on = 32e-3; % Ampères
I_TD1208R_off = 1.8e-6; % Ampères
U_TD1208R = 3; % Volt

P_TD1208R_on = U_TD1208R * I_TD1208R_on; % Watt
P_TD1208R_off = U_TD1208R * I_TD1208R_off; % Watt

P_TD1208R_moy = P_TD1208R_on * Rapport_on + ...
                P_TD1208R_off * Rapport_off; % Watt

```

STM32

```

% Calcul de la consommation de la carte. En se basant sur la datasheet.
% P_STM32_moy représente la puissance moyenne consommée par la carte en
% fonction de la durée où il est allumé et la durée où il est en DeepSleep.

I_STM32_on = 105e-3; % Ampères
I_STM32_off = 0.8e-6; % Ampères
U_STM32 = 3.6; % Volt

P_STM32_on = U_STM32 * I_STM32_on; % Watt
P_STM32_off = U_STM32 * I_STM32_off; % Watt

P_STM32_moy = P_STM32_on * Rapport_on + ...
              P_STM32_off * Rapport_off; % Watt

```

Batterie

```
% Modélisation de la batterie en se basant sur la datasheet. Condition
% initiale : batterie chargée

Q_batt = 1050e-3;                                % A.h
U_batt = 3.7;                                     % V
E_batt = 3.9;                                     % W.h
E_batt_t(1) = E_batt;
```

Chargeur

```
% Modélisation du chargeur en se basant sur la datasheet

I_charge = 500e-3;                                % Ampères
I_decharge = 1;                                   % Ampères
U_chargeur = 5;                                   % Volt

P_charge = U_chargeur * I_charge;                 % Watt
P_decharge = U_chargeur * I_decharge;             % Watt
```

Cellule solaire

```
% Modélisation du panneau solaire en se basant sur la datasheet.
% L'irradiance a été calculée à partir des données du site :
% http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php pour la ville de Paris
% le mois de Décembre. P_sol_max représente la puissance que peut fournir
% le panneau dans les meilleures conditions (1000 W/m^2), P_sol_max_h
% représente la puissance que pourra fournir le panneau dans les conditions
% les plus défavorables

I_sol = 540e-3;                                    % Ampères
U_sol = 5.5;                                       % Volt
P_sol_max = U_sol * I_sol;                         % Watt

Irradiance_max = 1000;                             % W/m^2
Irradiance_h = 40;                                % W/m^2

P_sol_max_h = P_sol_max * Irradiance_h / Irradiance_max; % Watt
```

Courbe solaire

```
% Modélisation de la courbe du soleil. Résolution de 3 equations à
% 3 inconnues grace à polyfit

x = [Sunrise*60 Midday*60 Sunset*60];
y = [0 1 0];
p = polyfit(x,y,2);
a = p(1);
b = p(2);
c = p(3);

% y = a * x^2 + b * x + c
```

Duree sur batterie

```
% Calcul pour chaque instant, la quantité d'énergie restante dans la
% batterie.

P_tot_decharge = P_STM32_moy + P_HCSR04_moy + P_TD1208R_moy + P_BMP_moy;
P_tot_charge = P_sol_max_h;

courbe_sol(t_max*60+1) = 0;

for cpt = t
    if cpt ~= 0
        if mod(cpt,24*60) > Sunrise*60 % apres 9h
            if mod(cpt,24*60) < Sunset*60 % avant 17h
                ensol(cpt+1) = 1;
                courbe_sol(cpt+1) = (a * mod((cpt+1),1440).^2 + ...
                                     b * mod((cpt+1),1440) + ...
                                     c) * ensol(cpt+1);
            else
                courbe_sol(cpt+1) = 0;
            end
        end
        if ensol(cpt) == 1 % si il y a du soleil
            E_batt_t(cpt+1) = E_batt_t(cpt) + ...
                             P_tot_charge * courbe_sol(cpt+1) / 60 - ...
                             P_tot_decharge / 60;
            if E_batt_t(cpt+1) > E_batt
                E_batt_t(cpt+1) = E_batt;
            end
        end
        if ensol(cpt) == 0 % si il n'y a pas du soleil
            E_batt_t(cpt+1) = E_batt_t(cpt) - P_tot_decharge / 60;
            if E_batt_t(cpt+1) < 0
                E_batt_t(cpt+1) = 0;
            end
        end
    end
end
end
```

Affichages

```
figure
XMIN = 0;
XMAX = 60*t_max;
YMIN = -0.1;
YMAX = 1.1;
subplot(2,1,1)
plot(t,courbe_sol)
ylabel('Courbe du soleil')
axis([XMIN XMAX YMIN YMAX])

YMIN = -0.1;
YMAX = 4;
subplot(2,1,2)
plot(t,E_batt_t)
ylabel('Energie dans la batterie (W.h)')
xlabel('temps (min)')
axis([XMIN XMAX YMIN YMAX])
```

