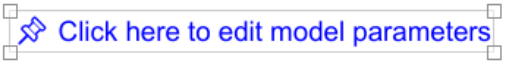
DL PEL-HIL Getting Started

Control Model Settings

DL R&D

2025-08-05

Place the mouse cursor over the ‘ Click here to edit model parameters’ field in the target model; it will turn blue. Click it.

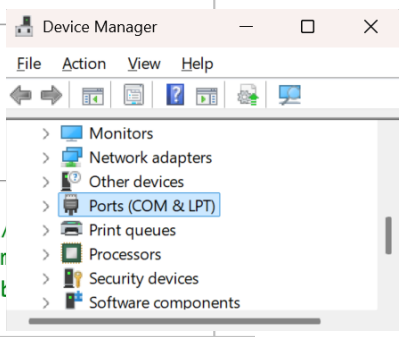


Basic Parameter Settings - *Sufficient for most projects*

In MATLAB Editor, users can edit the following parameters:

Parameters	Notes
PWM_frequency	[Hz], switching frequency of the power converter.
deadtime	[s], a rising edge delay inserted between switching the two IGBTs of the same bridge leg of the power converter to prevent them from conducting simultaneously and causing a short circuit.
Ts	[s], sample time for the control system, usually set to the reciprocal of the PWM frequency.
target.comport	The COM port for DL RCPCORE recognized on the PC, which can be found in the Device Manager.

```
% Copyright 2025 De Lorenzo Spa

%% initiation
close all;
clearvars;
clc;

%% Set PWM Switching frequency
PWM_frequency    = 4e3;          %Hz          // Converter s
T_pwm            = 1/PWM_frequency;  %s  // PWM switching tim
deadtime         = 5e-6;         %s //rising edge delay for b

%% Model Parameters
Ts               = T_pwm;        %s      // Sample time for control system
dataType         = 'single';            % Floating point

%% Hareware parameters
% Set controller parameters
target           = SetControllerDetails('DL RCPCORE', PWM_frequency, deadtime);
target.comport   = 'COM6';       % Update the appropriate serial port
converter        = SetConverterParameters('DL 2106T06');

disp(target);
disp(converter);
```



Advanced Parameter Settings – *for deep customization*

DL RCPCORE Parameters

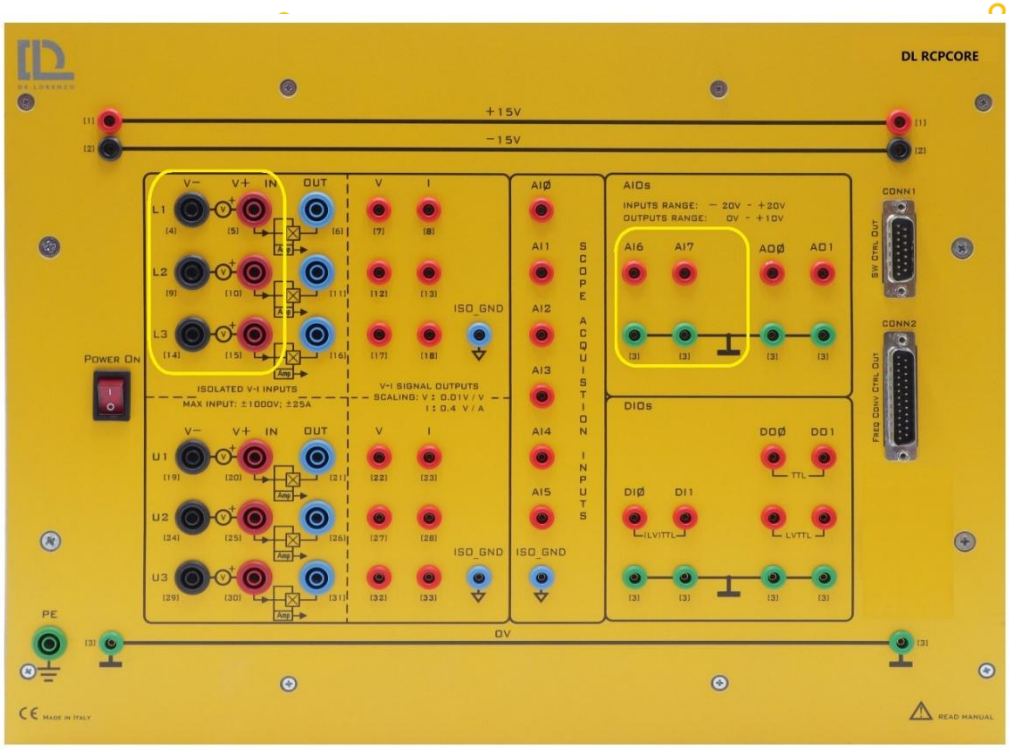
Find the '**SetControllerDetails**' function in the script, place the cursor inside the function, and then press **Ctrl+D** or right-click and select **Open "SetControllerDetails"** to edit the function.

The parameters that can be modified in the **case {'DL RCPCORE'}** are as follows:

```
case {'DL RCPCORE'}
    target.model                = 'DL RCPCORE'; %       // RCP (Rapid Control Prototyping) Module
    target.CPU_frequency        = 100e6;                %Hz     // Clock frequency
    target.PWM_frequency        = PWM_frequency;        %Hz     // PWM frequency
    target.PWM_Counter_Period   = round(target.CPU_frequency/target.PWM_frequency/2); % //PWM timer counts for up-down counter
    target.PWM_Counter_Period   = target.PWM_Counter_Period+mod(target.PWM_Counter_Period,2); % // Count value needs to be even
    target.Half_PWM_Counter_Period = single(target.PWM_Counter_Period*0.5);
    target.ADC_Vref             = 3;                    %V      // ADC voltage reference
    target.ADC_MaxCount         = 4095;                 %       // Max count for 12 bit ADC
    target.SCI_baud_rate        = 12e6;                 %Hz     // Set baud rate for serial communication
    target.GainV = 0.4883;
    target.GainI = 0.0122;
    target.GainAI6_7 = 0.0097656;
    target.Offset_VL1 = 0;      %V Offset of the VL1 high-voltage measurement channel when measuring 0V.
    target.Offset_VL2 = 0;      %V Offset of the VL2 high-voltage measurement channel when measuring 0V.
    target.Offset_VL3 = 0;      %V Offset of the VL3 high-voltage measurement channel when measuring 0V.
    target.Offset_AI6 = 0;      %V Offset of the AI6 low-voltage measurement channel when measuring 0V.
    target.Offset_AI7 = 0;      %V Offset of the AI7 low-voltage measurement channel when measuring 0V.
```

target.SCI_baud_rate specifies the baud rate for real-time communication between the target and host models.

target.Offset_x represent the hardware offsets of the corresponding high-voltage and low-voltage measurement channels in the DL RCPCORE at 0V, used for static hardware calibration.



DL 2106T06 Parameters

Find the '**SetConverterParameters**' function in the script, place the cursor inside the function, and then press **Ctrl+D** or right-click and select **Open " SetConverterParameters"** to edit the function.

The parameters that can be modified in the **case {'DL 2106T06'}** are as follows:

```
case 'DL 2106T06'
    converter.model             = 'DL 2106T06';     % Frequency converter
    converter.ADCnGain          = 0.9768;           % gain for conversion from 12-bit to n (rpm)
    converter.ADCIinvGain       = 2.8571e-3;        % gain for conversion from 12-bit to Iinv (A)
    converter.ADCVdcGain        = 0.1114;           % gain for conversion from 12-bit to Vdc (V)
    converter.nOffset         = single(508);        %Counts // ADC Offset for n
    converter.IinvOffset      = single(0.03);       %Counts // ADC Offset for Iinv
    converter.VdcOffset       = single(2.7);        %Counts // ADC Offset for Vdc
```

converter.xOffset represent the hardware offsets of the corresponding analogue signal measurement channels in the DL 2106T06 at 0V, used for static hardware calibration.