PlanetCNC Wireless handwheel user manual

Supported features:

-Wireless communication

-No additional software plug-ins needed

-Display with backlight showing work, machine coordinates and override

-All 9 axes supported

-5 selectable jogging modes

-Spindle override control

-Feed override control

-Machine control buttons (E-Stop, Start, Pause…)

-Button function customization

-Offset control buttons

-Spindle and coolant control buttons

-Durable rubber protection guard

-Long lasting batteries (Batteries are not included!)

-High quality long range easy mountable antenna with detachable cable

Kit includes:

-Wireless handwheel with rubber guard

-Antenna with extension cable

-USB dongle transceiver

-Metal hanger bracket and screws for mounting

-Requires 2 x AA batteries (Batteries are not included!)

Startup process:

-Insert 2 x AA batteries

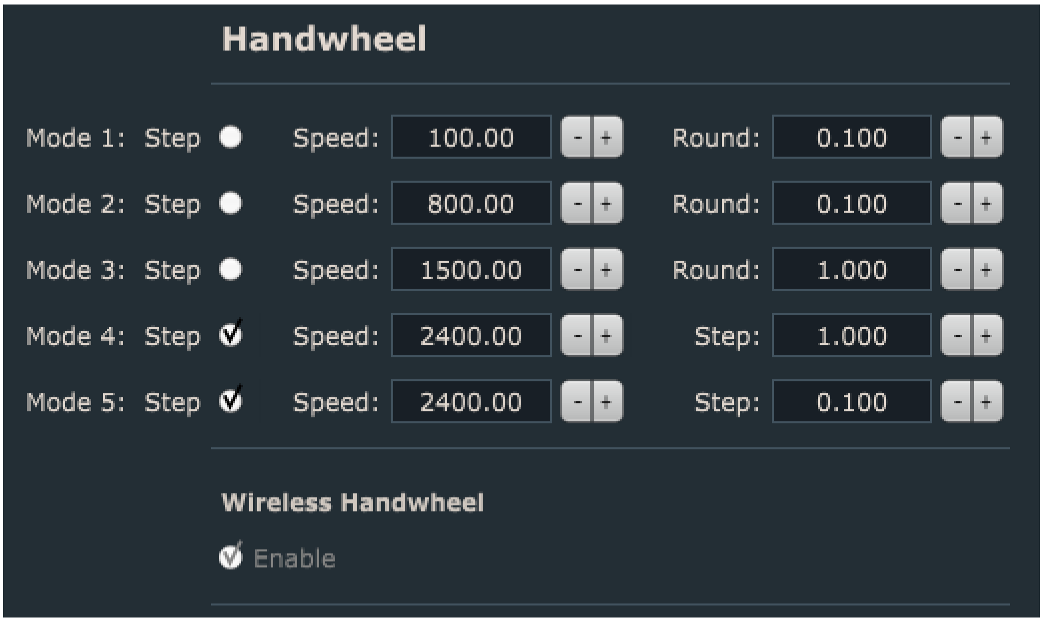
-Attach cable with antenna and USB dongle.

-Connect the USB dongle with your PC's USB port.

-Turn ON the handwheel using ON/OFF switch

-Launch PlanetCNC TNG sw and check the Jogging/Handwheel settings section.

If handwheel is recognized by TNG software, *Wireless Handwheel* option should be enabled:



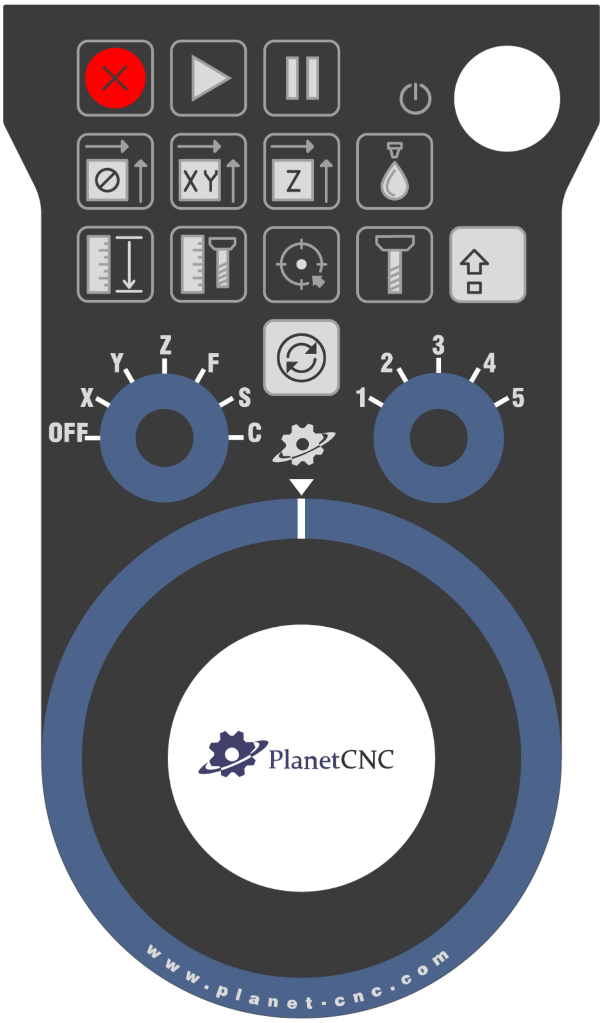
Handwheel

Mode 1:	Step ○	Speed:	100.00	- +	Round:	0.100	- +
Mode 2:	Step ○	Speed:	800.00	- +	Round:	0.100	- +
Mode 3:	Step ○	Speed:	1500.00	- +	Round:	1.000	- +
Mode 4:	Step ✓	Speed:	2400.00	- +	Step:	1.000	- +
Mode 5:	Step ✓	Speed:	2400.00	- +	Step:	0.100	- +

Wireless Handwheel

✓ Enable

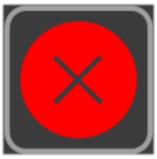
Handwheel button layout:



Handwheel button functionality:

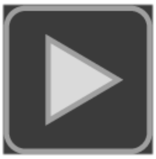
E-Stop button:

Initiates E-Stop.



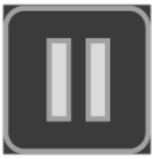
Start:

Starts program.



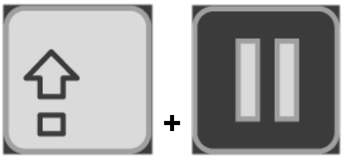
Pause:

Pauses program.



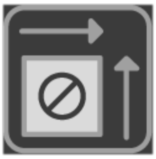
Stop:

Stops program.



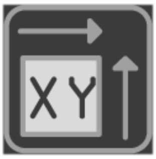
Offset zero button:

Sets work position offset value to zero.



Work position XY zero button:

Sets current XY work position to zero.



Work position Z zero button:

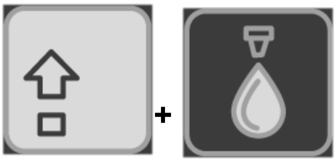
Sets current Z work position to zero.

Flood ON/OFF button:

Turns flood ON/OFF.

Mist ON/OFF button:

Turns mist ON/OFF.



Measure Height button:

Initiates Measure Height procedure.

Measure Tool Length button:

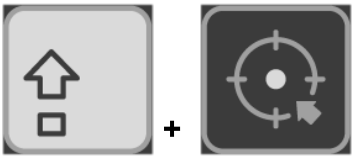
Initiates Tool Length measurement procedure.

Move To Zero XY button:

Initiates Move to Zero procedure.

Home button:

Initiates Home procedure.



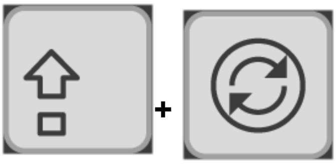
Spindle ON/OFF button:

Turns spindle ON/OFF.

Cycle button with axis knob selected:

Toggles Machine and Work coordinates.

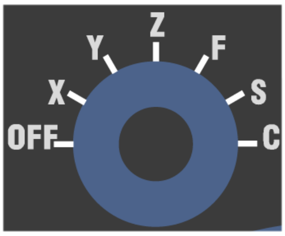
Selects group of axes: X,Y,Z or A,B,C or U,V,W



Cycle button with feed override know selected:

Toggles Feed and Traverse override.

Axis, Feed override, Spindle override and Info select switch:



X,Y,Z (A,B,C,U,V,W): Axis selected will move when encoder wheel is rotated.

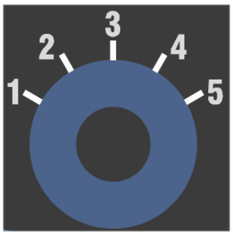
F: Feed speed and traverse override menu.

S: Spindle speed override menu.

C: Additional info of currently selected jogging mode or macro selection (see button customization chapter)

Jogging mode select switch:

Selects jogging mode from 1-5. Jogging modes are configured in PlanetCNC TNG sw under File/Settings/Jogging/Handwheel. See PlanetCNC TNG user maual for more info.



Button function customization:

User can customize handwheel button functionality. This can be done by configuring the ***Handwheel.txt*** file.

First, create a blank *Handwheel.txt* file and place it in your profile root folder.

How to configure this file will be described further in this document.

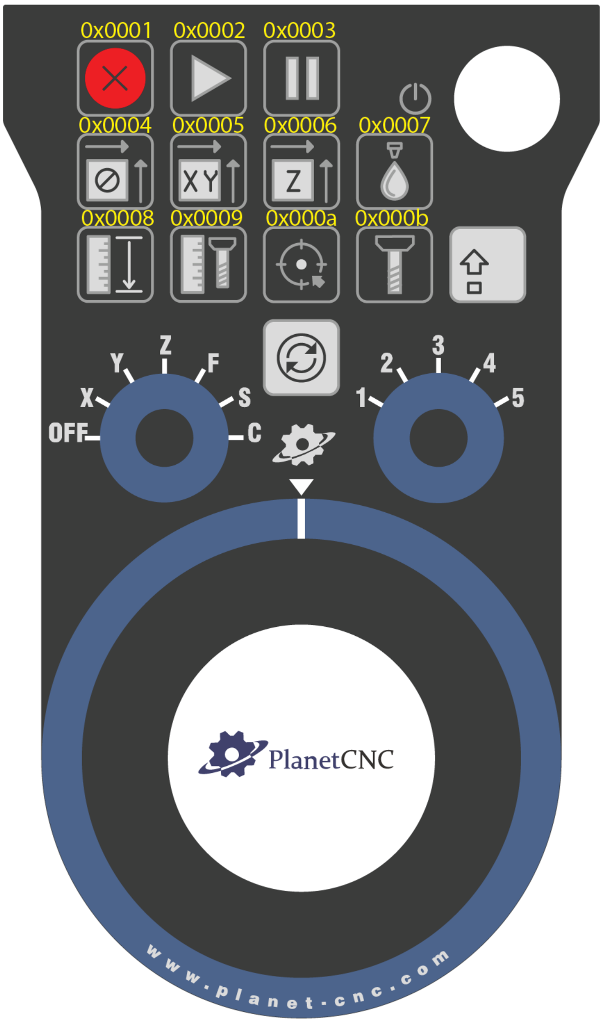
Button HEX values:

Each button of the handwheel keypad is represented by its default HEX value. These values could also be in a decimal form, but HEX gives better and more intuitive description.

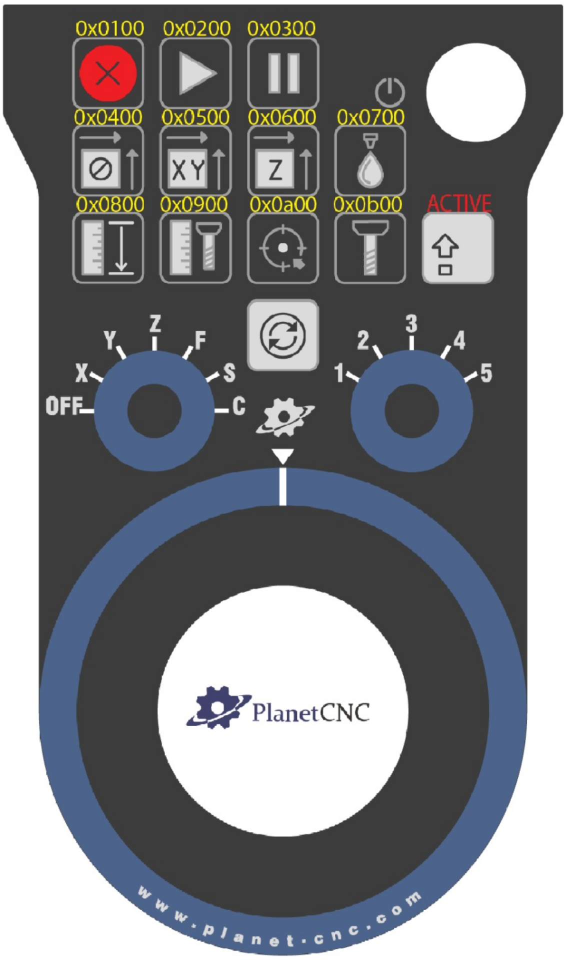
Each button can be represented with two types of HEX values, *Lower HEX* (SHIFT button is not active) and *Upper HEX* values (SHIFT button is active).

Best to describe it is by using illustrated view:

Lower HEX button values (SHIFT button not active):



While holding SHIFT button, each buttons is now represented with the Upper HEX value:



Now that we know LOWER and UPPER values of each button, we can easily assign new alternative functions of the handwheel buttons.

Button	Lower HEX value	Upper HEX value
E-Stop	0x0001	0x0100
Start	0x0002	0x0200
Pause	0x0003	0x0300
Offset zero	0x0004	0x0400
Work position XY zero	0x0005	0x0500
Work position Z zero	0x0006	0x0600
Flood	0x0007	0x0700
Measure Height	0x0008	0x0800
Measure Tool Length	0x0009	0x0900
Move To Zero XY button:	0x000a	0x0a00
Spindle ON/OFF	0x000b	0x0b00

BASIC Handwheel.txt file configuration:

Earlier chapter can be better described with the actual command lines of the configuration file.

Below are a default* program functions that will be executed when corresponding button will be activated:

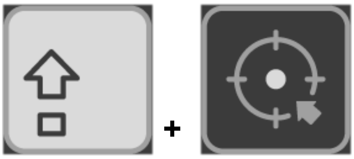
```
cmd: "Machine.Emergency_Stop" "" num=0x0001
cmd: "Machine.Start" "" num=0x0002
cmd: "Machine.Pause" "" num=0x0003
cmd: "Machine.Work_Position.Offset.To_Zero" "" num=0x0004
cmd: "Machine.Work_Position.Axis_To_Zero.XY" "" num=0x0005
cmd: "Machine.Work_Position.Axis_To_Zero.Z" "" num=0x0006
cmd: "Machine.Flood" "" num=0x0007
cmd: "Machine.Work_Position.Measure_Height" "" num=0x0008
cmd: "Machine.Tool_Offset.Measure_Length" "" num=0x0009
cmd: "Machine.Move.Axis_To_Zero.XY" "" num=0x000a
cmd: "Machine.Spindle" "" num=0x000b
```

*(Note that this is a default behaviour of handwheel buttons. No configuration file or configuration of it is needed to obtain it…)

Lines below will execute Homing procedure, Machine Stop and Mist ON/OFF actions. Becuse upper HEX values are used, this indicates that SHIFT button needs to be pressed:

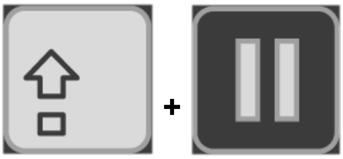
Homing procedure:

cmd: "Machine.Home" "" num=0x0a00



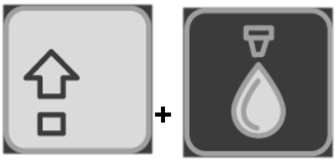
Machine Stop:

cmd: "Machine.Stop" "" num=0x0300



Mist On/Off:

cmd: "Machine.Mist" "" num=0x0700



So if user wishes to add an alternative functionality to a button, all it needs to be done is to add a line which would execute desired program function. Upper HEX value of button should be used.

Example:

We would like to execute *Surface Measure* procedure when we press the Start button in combination with SHIFT button.

Since we will assign an alternative button functionality we need to obtain the Upper HEX button value. Earlier chapter with the layout illustration gives this information → 0x0200

Cmd line is usually the menu path of the program function. In this case this would be:

Machine/Measure/Surface

Note that path menu levels are represented with “.” and spaces in the program function name with “_” character.

Command line that would help us achieve this is:

```
cmd: "Machine.Measure.Surface" "" num=0x0200
```

If we would like to turn ON/OFF Output 1 with the same button instead, we would write:

```
cmd: “Machine.Outputs.Output_1” "" num=0x0200
```

ADVANCED Handwheel.txt file configuration:

Since there are not enough buttons for every possible function of tool that TNG offers, macros can be created. Macros are able to execute program functions, expression functions, parameter manipulation etc.

They can be navigated and used trough the Custom menu.

How to create a macro:

Just as we added command lines for button alternative functions to Handwheel.txt file, in the same way we add dedicated lines to create macros.

We will demonstrate three examples of macro use. Macro for program function, program function with parameter manipulation and expression function.

Macro command line for program function File Open:

```
cmd: "File.Open_..." "Open file" num=0x10001
```

cmd: "File.Open_..."
Like before, command line is the path of the program function, File/Open

"Open file"
Macro will be displayed under this name in the custom menu.

num=0x10001
This num value serves as a macro ID as also as a menu sequence number.
First value 1 identifies this as a macro line, and last value 1 is its sequence number.

Macro command line for expression function with parameter manipulation:

```
cmd: "Machine.Output_PWM" "PWM" num=0x10002 param=0|10|0|100 val="[0;expr:_hw_mpg_custom]"
```

cmd: "Machine.Output_PWM"
Like before, command line is the path of the program function, Machine/Output_PWM

"PWM"
Macro bill be displayed under this name in the custom menu.

num=0x10002
This num value serves as a macro ID as also as a menu sequence number.
First value 1 identifies this as a macro line, and last value 2 is its sequence number.

param=0|10|0|100
These are definition values of PWM signal that will be changed trough this macro.

So, when the PWM signal will be modified, if will consider these definitions:
0[initial value] | 10[increment value] | 0[min value] | 100[max value]

val="[0;expr:_hw_mpg_custom]"
0 is a number of output pin that will generate the PWM signal. Zero based numbering is used (1st pin has number zero, 2nd pin has number one etc…)
Macro uses parameter *_hw_mpg_custom* for PWM signal value modification.

Macro command line for expression function:
This macro executes expression function located in the Expr.txt file.

```
expr: "exec('#Message')" "Message" num=0x10003
```

expr: "exec('#Message')"
Since *exec* is an expression function, we need to use *expr* command at the beginning of the macro line. *exec* command will execute the #Message function located in our Expr.txt file.

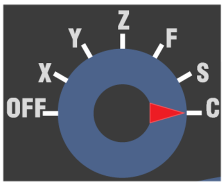
In this case, *#Message* function looks like this:
#Message
exec(msg('Handwheel function trigger'));

“Message”
Macro bill be displayed under this name in the custom menu.

num=0x10003
This num value serves as a macro ID as also as a menu sequence number.
First value 1 identifies this as a macro line, and last value 3 is a sequence number.

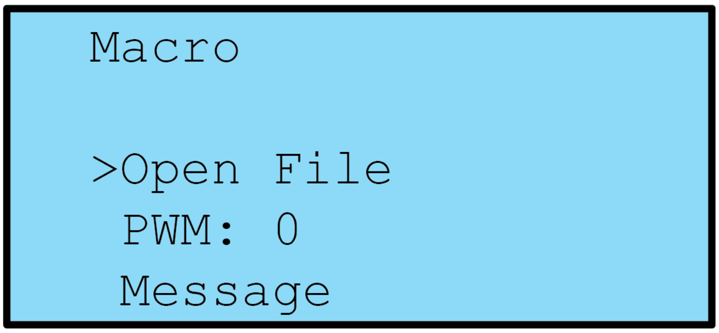
Accessing and using the Custom menu

Custom menu can be accessed using axis switch knob set at C position:



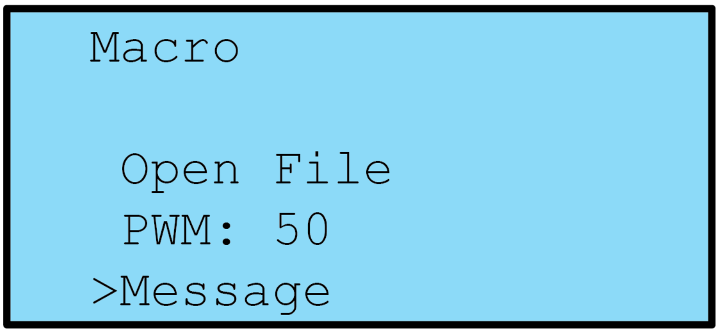
At first, display will show current jogging mode info. If the handwheel encoder is rotated custom menu will be displayed.

As per our configuration, three macros will be available for selection. Using a handwheel encoder, user can navigate trough the menu items:



```
Macro

>Open File
 PWM: 0
 Message
```

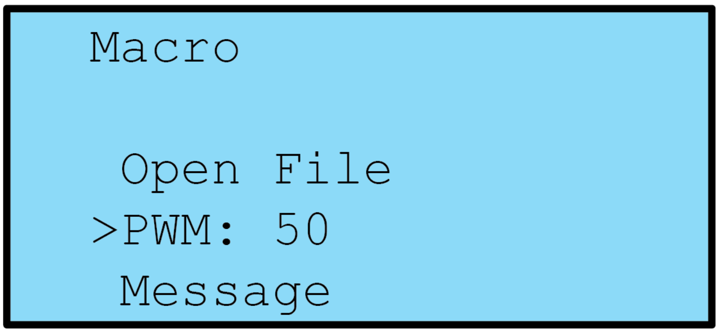


```
Macro

 Open File
 PWM: 50
>Message
```

To execute the *Open File* or *Message* macro, macro should be selected in the menu and confirmed with Cycle button: 🔄

To execute *PWM* macro, we select it in the menu and by simultaneously using a Shift button and an encoder wheel we can change the PWM duty cycle value:



```
 Macro

  Open File
 >PWM: 50
  Message
```

To actually set PWM on the output pin, we just need to confirm it with the Cycle button:

🔄