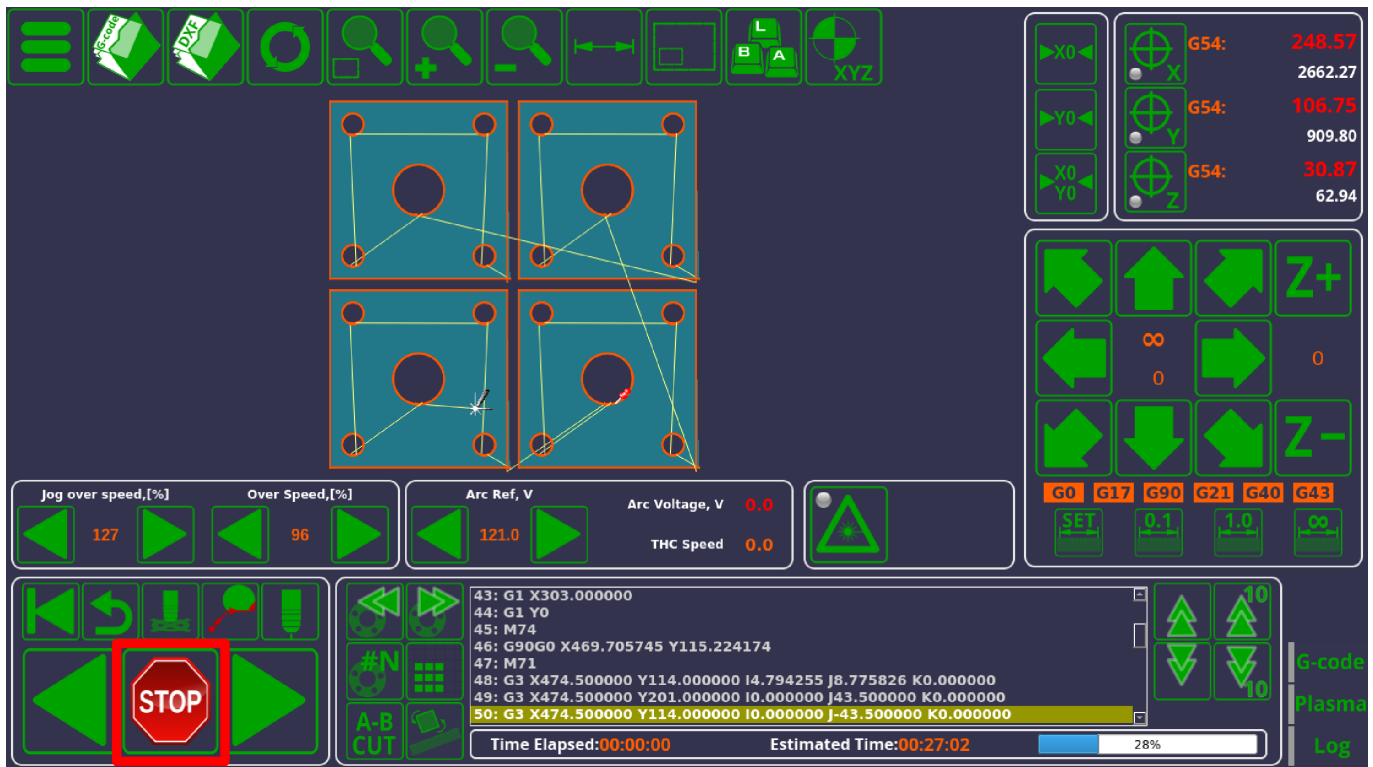# Stop/End program button commands

When using myCNC, the actions performed by the program when the Stop button is pressed are defined in the following files:

- M02.plc (Settings > Config > PLC > Hardware PLC)
- OFF.plc (Settings > Config > PLC > Hardware PLC)
- __HANDLER_GCODE_STOP (Settings > Config > PLC > Software PLC)

The actions taken upon pressing the button differ depending on the sequence of events. If pressing ONCE, the M02.plc procedure is started (below is the M02.plc for the Plasma profile X1366P):

M02.plc

```
#include pins.h
#include vars.h
#include func_plasma.h

main()
{
  timer=0;

  portclr(OUTPUT_DRILL_VALVE_DOWN);
  portset(OUTPUT_DRILL_VALVE_UP);
  portclr(OUTPUT_DRILL_POWER);

  stop_thc();
  stop_trigger1();
```

```
  //turn off power source
  portclr(OUTPUT_PLASMA);

  portclr(OUTPUT_PROBE);
  test_lift_after_cut();
  do_lift_after_cut();
  proc=plc_proc_idle;

  start_trigger2();
  exit(99);
};
```

If the Stop button is pressed twice, the OFF.plc procedure will be utilized:

OFF.plc

```
#include pins.h
#include vars.h
#include func_plasma.h

main()
{

  portclr(OUTPUT_DRILL_VALVE_DOWN);
  portset(OUTPUT_DRILL_VALVE_UP);
  portclr(OUTPUT_DRILL_POWER);


  //stop_thc();
  //stop_trigger1();

  //turn off power source
  portclr(OUTPUT_PLASMA);
  portclr(OUTPUT_PROBE);
  //proc=plc_proc_idle;

  exit(99);
};
```

As can be seen, while the M02.plc procedure will have some sort of lift control to prevent issues do to the working tool potentially coming into contact with the working material/surface, the OFF.plc command will simply stop all movement, clear all relevant ports to turn the system off, and end the program. Sometimes, when issues arise with the lift procedure, or with any of the processes described in the M02 procedure, it is advisable to click the Stop button twice in order to obtain a complete and total G-code run shutdown.

Additionally, whenever the player stops running G-code, the __HANDLER_GCODE_STOP Software PLC which is continuously running in the background will be activated. This handler (along with the other Software PLC handlers) allows to monitor for some event, such as stopping a G-code run, and having

some code be executed after the event has occurred. By default, the __HANDLER_GCODE_STOP simply consists of the following code:

__HANDLER_GCODE_STOP

```
main()
 {
  exit(99);
 };
```

Therefore, in a default configuration, this handler will simply activate, do nothing and then close. However, in situations where it is necessary to add a command which will activate whenever the G-code run has stopped, __HANDLER_GCODE_STOP can be used. For example, the code can be altered to look the following way if it is necessary, for example, to turn off the output port #3 when the run is stopped:

__HANDLER_GCODE_STOP

```
main()
 {
  portclr(3);
  exit(99);
 };
```

In non-default configurations, there can be a number of outputs/inputs being set and cleared upon program stop. Often, when issues arise with the ports being set and cleared seemingly at random upon program start/stop, it is an issue with code responsible for the port set / port clear being left over in the M02, OFF, and __HANDLER_GCODE_STOP commands.