

ERCF v2 - HAPPY HARE SUPPLEMENT

The happy hare, with each joyful leap, reminds us that life's truest richness lies not in the speed of our achievements, but in savoring the moments along the journey. 🐰

INSTALL MMU FIRMWARE

ERCF v2 uses Happy Hare firmware for controlling MMU functions under Klipper.

Download from the github repo and run the installer.

Note: if you have a non standard Klipper installation pay attention to the '-c' and '-k' options

```
cd ~
git clone https://github.com/moggieuk/Happy-Hare.git
cd Happy-Hare

./install.sh -i
```

```
Klipper service found
Reading default configuration parameters...

-----
Let me see if I can get you started with initial configuration
You will still have some manual editing to perform but I will explain that later
(Note that all this script does is set a lot of the time consuming parameters in the config

What type of MMU are you running?
1) ERCF v1.1 (inc TripleDecky, Springy, Binky mods)
2) ERCF v2.0 (inc ThumperBlocks mod)
3) Tradrack v1.0
4) Other (Custom creations or variations not mentioned above...)
MMU Type? (1-4)? 2
Some popular upgrade options for ERCF v2.0 can automatically be setup. Let me ask you about them..
Are you using 'ThumperBlocks' filament block option (y/n)? n

-----
Select mcu board type used to control MMU
1) BTT MMB
2) Fysetc Burrows ERB
3) Standard EASY-BRD (with SAMD21)
4) EASY-BRD with RP2040
5) Not in list / Unknown
MCU type? (1-5)? 1
```

RUN THROUGH INTERVIEW

Make sure you select ERCF v2.0 and correct MCU board (BTT MMB recommended)

If you are not sure of options, decline them at this stage – you can configure them later

Answer 'no' to ERCF compatibility



FOLLOW HAPPY HARE SETUP GUIDELINES

A ERCF v2 landing page had been created to guide you through the essential hardware verification and setup process.

The basic steps are:

- Valid klipper configuration and control
- Run the sequence of calibration routines
- Fine tune options

Landing Page: https://github.com/moggieuk/Happy-Hare/tree/main/doc/ercf_v2.md

BASIC MMU COMMANDS

Happy Hare has a rich command set and extensive documentation but here is a quick reference of the basic commands

```
Happy Hare MMU commands: (use MMU_HELP MACROS=1 TESTING=1 for full command set)
MMU : Enable/Disable functionality and reset state
MMU_CHANGE_TOOL : Perform a tool swap
MMU_CHECK_GATE : Automatically inspects gate(s), parks filament and marks availability
MMU_STATS : Dump or reset the MMU statistics
MMU_EJECT : Eject filament and park it in the MMU or optionally unloads just the extruder (EXTRUDER_ONLY=1)
MMU_ENCODER : Display encoder position or temporarily enable/disable detection logic in encoder
MMU_ENDLESS_SPOOL : Display or redefine the EndlessSpool groups
MMU_FORM_TIP : Convenience macro to call the standalone tip forming functionality
MMU_HELP : Display the complete set of MMU commands and function
MMU_HOME : Home the MMU selector
MMU_LED : Manage mode of operation of optional MMU LED's
MMU_LOAD : Loads filament on current tool/gate or optionally loads just the extruder for bypass or recovery usage
MMU_MOTORS_OFF : Turn off both MMU motors
MMU_PAUSE : Pause the current print and lock the MMU operations
MMU_PRELOAD : Preloads filament at specified or current gate
MMU_PRINT_END : Restore MMU idle state after print
MMU_PRINT_START : Initialize MMU state and ready for print
MMU_RECOVER : Recover the filament location and set MMU state after manual intervention/movement
MMU_REMAP_TTG : Display or remap a tool to a specific gate and set gate availability
MMU_RESET : Forget persisted state and re-initialize defaults
MMU_SELECT : Select the specified logical tool (following TTG map) or physical gate
MMU_SELECT_BYPASS : Select the filament bypass
MMU_SERVO : Move MMU servo to position specified position or angle
MMU_GATE_MAP : Display or define the type and color of filaments on each gate and optionally spoolman ID
MMU_STATUS : Complete dump of current MMU state and important configuration
MMU_SYNC_GEAR_MOTOR : Sync the MMU gear motor to the extruder motor
MMU_TOOL_OVERRIDES : Displays, sets or clears tool speed and extrusion factors (M220 & M221)
MMU_UNLOCK : Wakeup the MMU prior to resume to restore temperatures and timeouts
```

KLIPPER CONFIG FILE CHANGES

In addition to the '[include]' entries created by Happy Hare, there are a few changes that you must make to your klipper config in order to support an MMU:

Add to your '[extruder]' section:

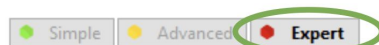
```
[extruder]
max_extrude_cross_section: 50      # MMU needs this
max_extrude_only_distance: 200    # MMU needs this
```

Make sure '[virtual_sdcard]' section exists similar to:

```
[virtual_sdcard]
path: ~/printer_data/gcodes
on_error_gcode: CANCEL_PRINT
```

This part of the manual will guide you through setting up your slicer to work with the ERCF. This guide assumes you are using SuperSlicer but it will most likely work the same with PrusaSlicer. Only the changes from a default (i.e. single extruder and single tool) profile will be discussed.

Cura and other slicers will not be covered.

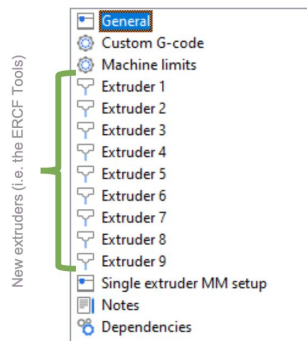
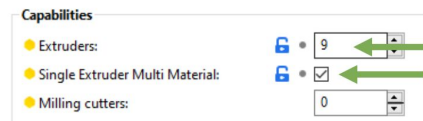


Make sure you have access to all fields

GENERAL

In the general section of the Printer Settings Tab, activate the 'Single Extruder Multi Material' option and set the proper number of 'Extruders' (i.e. the number of channels on your ERCF)

The different extruders will then appear in the menu list



EXTRUDER N

For each of the Extruder in your list, make sure that the 'Tool name' is left blank



CUSTOM G-CODE : TOOL CHANGE

Set the 'Tool change G-code' to 'T[next_extruder]', this will make the Gcode generate 'T0' Gcode to request a Tool Change for Tool 0 for example ('T1' for Tool 1 etc.)

Tool change G-code



T[next_extruder]

GLOBAL BEHAVIOR

The Start and End gcode showed below are designed for using the ERCF as the filament management solution, even for single color prints. This means that the proper filament will be loaded at the beginning of the print (right after the print_start macro has finished) and the filament will be unloaded at the end of the print. So, when your printer is not printing, no filament will be in the toolhead and the whole print_start gcode will be executed without any filament loaded, so adapt it accordingly (e.g. if you have any purge sequence in it, that will be useless)

CUSTOM G-CODE : PRINT START

By default slicers do not load the first tool of a print, therefore you can either opt to always load the first tool prior to starting the print or else you need to pass the initial tool to your print start macro. This example also sets another variable with a list of tools used in a print (see START_PRINT gcode for usage):

Start G-code



```
M190 S0 ; removes setwait temp gcode added automatically
M109 S0 ; removes setwait temp gcode added automatically
START_PRINT INITIAL_TOOL={initial_tool} REFERENCED_TOOLS={referenced_tools}
```

```
[gcode_macro START_PRINT]
```

description: Called when starting print

gcode:

```
{% set REFERENCED_TOOLS = params.REFERENCED_TOOLS|default("")|string %}
{% set INITIAL_TOOL = params.INITIAL_TOOL|default(0)|int %}
```

```
{% if REFERENCED_TOOLS == "!referenced_tools!" %}
  RESPOND MSG="Happy Hare gcode pre-processor is disabled"
  {% set REFERENCED_TOOLS = INITIAL_TOOL %}
{% elif REFERENCED_TOOLS == "" %}
  RESPOND MSG="Happy Hare single color print"
  {% set REFERENCED_TOOLS = INITIAL_TOOL %}
{% endif %}
```

: <your logic here>

Example inclusion of logic to check that all tools used by the print are available and to load the initial tool

```
MMU_CHECK_GATE TOOLS={TOOLS_USED}
MMU_CHANGE_TOOL STANDALONE=1 TOOL={INITIAL_TOOL}
```

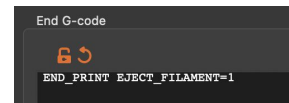
Read the parameters passed in by custom g-code in slicer

Optional: Run Happy Hare function to check that all tools are available

Load the initial tool used in the print

CUSTOM G-CODE : PRINT END

You might prefer to unload the filament at the end of a print. To support this, add the following to your custom end g-code in your slicer:



```
[gcode_macro END_PRINT]

description: Called when ending print
gcode:
  {% set EJECT_FILAMENT = params.EJECT_FILAMENT|default(0)|int %}

  {% if EJECT_FILAMENT|int == 1 %}
    MMU_EJECT
  {% endif %}
Example inclusion of logic to optionally eject filament after print
```

Parameter passed in from slicer

Filament ejection command

TIP FORMING OPTIONS

There are three basic methods of dealing with filament tip creation:

1. Let the slicer do it in print (Happy Hare will do it when not in print)
2. Always let Happy Hare do it
3. Use filament cutter like ERF Filamatrix to avoid "tip tuning" altogether

Without any changes, option 1 above will be the default. It however is not recommended, instead use option 2 or 3.

TURNING OFF SLICER TIP CREATION (RECOMMENDED)

For both options 2 & 3 above it will be necessary to disable tip forming in your slicer. This will differ from slicer to slicer but in general you need to zero out these settings.

To enable option 2, ensure the following two parameters are set in Happy Hare 'mmu_parameters.cfg':

force_form_tip_standalone: 1

form_tip_macro: _MMU_FORM_TIP_STANDALONE

To enable option 3:

.force_form_tip_standalone: 1

form_tip_macro: _MMU_CUT_TIP

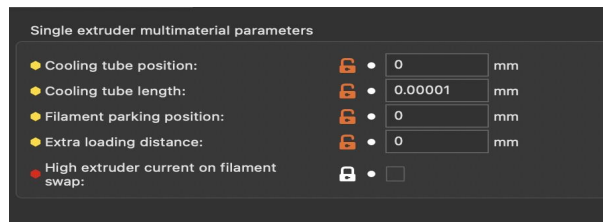
SLICER TIP FORMING (NOT RECOMMENDED)

For option 1 you have to both setup tip forming in the slicer and in the Happy Hare firmware for use when not printing. To do this edit the slicer panel similar to this but with settings tuned to you specific toolhead:

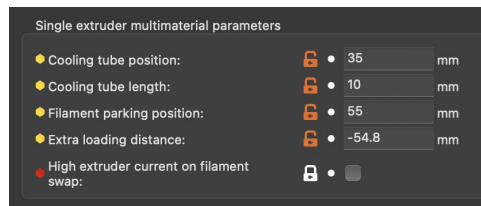
Also ensure the following two parameters are set in Happy Hare 'mmu_parameters.cfg':

force_form_tip_standalone: 0

form_tip_macro: _MMU_FORM_TIP_STANDALONE



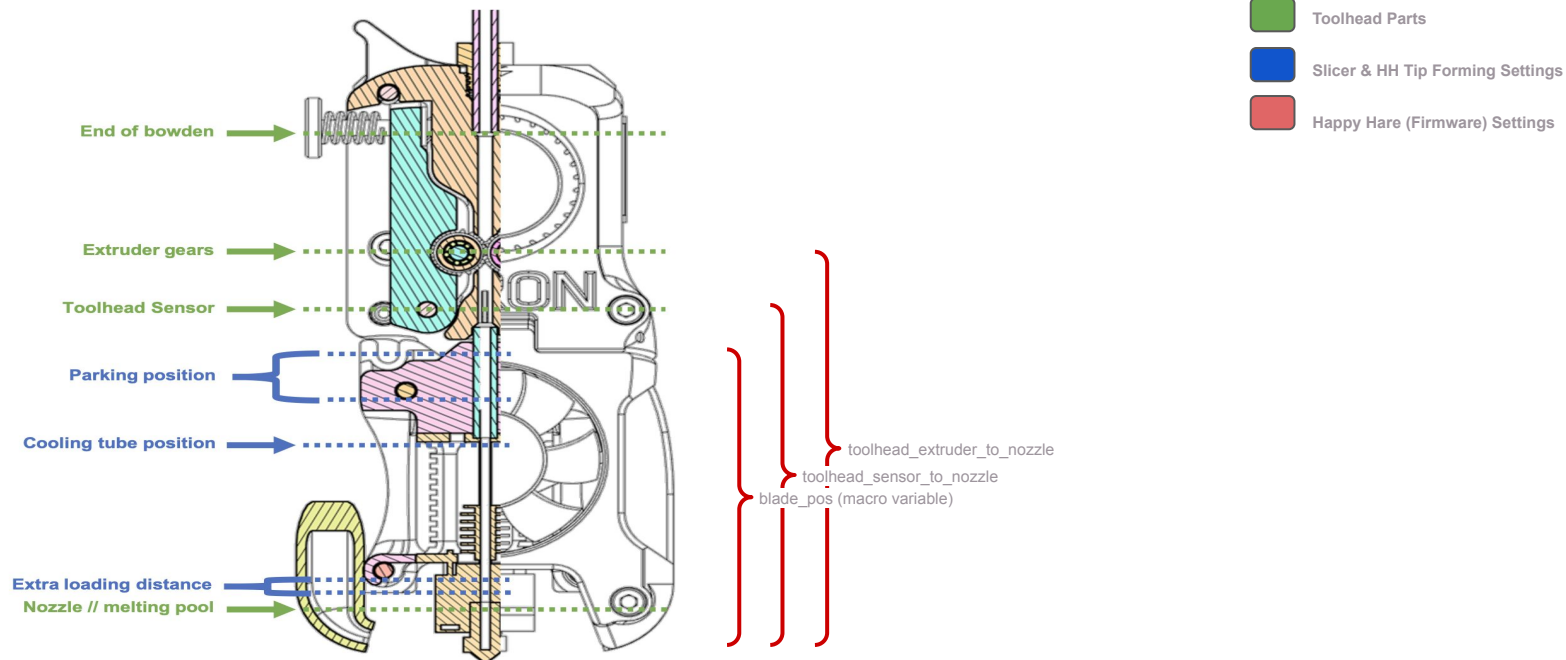
Note: A bug in SuperSlicer requires that one setting is near zero but not zero!



Consult your slicer documentation for setting these values

REFERENCE TOOLHEAD SCHEMATICS

Important reference for dimensions in your toolhead.



More doc to come

Please refer to the general Happy Hare documentation for more specific details