

- ProgrammedMotorMove function: parameters set from State to appropriate variable for appropriate move
- ContinousMotorMove function: either from user button press (Purge), or internal command variable (Compress until motor & encoder desync steps)

STOPS:
EMERGENCY STOP, if pressed, immediate stop of actual function. if function is NOT plunger down move (compression or injection), then releasing E-STOP will return to same point, if IS down move, then will immediately de Release Function, and then enter Purge Function so plunger can be moved manually

Endstops:
TopEndStop: when plunger reaches Top of stroke, the barrel is supposedly empty, so simply stop any more movement up (possible send plunger to Refill position, as is a nearby offset, or only allow move down*)

BottomEndStop: the encoder data should update the total stroke of plunger and add the NextActualInjectionParams to the actual plunger position, and advise if the injection can be completed or not, but if the BES is triggered without this warning, stop motor move, only allow motor move up*

Barrel Endstop: if the barrel has been detected as having moved from its fixed place, then stop any movement/ injection and advise with error, only allow up move to allow release of pressure so clamp can be inspected without pressure*

* in theory, in none of these cases coincide with a user button config that ALLOWS any movement anyway, so possibly could include this movement ONLY... also, these types of endstops errors can easily be indicated by flashing LEDs red in order:- if top endstop is triggered, flash goes up, bottom, flashing goes down, if barrel, centre and exterior alternate 😊



Press both bottom buttons 2s to change from black (!EndDay) to Blue (EndDay, no more Refills) or visa versa

Communications via UART:
During certain States, the STM32 will be listening for commands to change to other States, or to change ActualInjectParams, or CommonInjectParams... this will only happen during States that make sense, for example ReadyToInject, Refill, NOT during motorMoves

however, the STM32 will be transmitting every 1s to the ESP32 the actual Encoder position, the actual Nozzle temp, which the ESP32 display will later show a graphic rep of the barrel and the contents as rectangles, with either timers in each rectangle, or colours to show plastic melt stage... other good Encoder transmission points may be at the end of Refill/ Compression functions, as these will be the boundary between rectangular blocks representing plastic Refills and therefore melt stages

Also, on listening and receiving Params changes (for Common or Actual InjectParams, the STM32, after updating these variables, will transmit back to the Display so they can be shown on screen and compared to requested data

Buttons input and output:
Lower keypad consists of 3 buttons, select, up & down, with LEDs behind, that change use as per the State of the machine and user input required to operate.
Simple direct connection GPIO with Pullup is considered enough.

Barrel & Nozzle block detection however is still complicated, as the force experimented in this area (full downforce from Nema motor via 20:1 reductor, about 1500kg estimated min) makes use of traditional mechanical endstops very difficult, and proximity sensores also may not fully verify that a decent blocking mech is in place, contacting both against the platform and against the barrel / nozzle edge (as per compression or purge) - it is considered a L-shaped 30x3mm profile with a blind end, as used with the manual machine, together with a toggle mech to be able to invert this L-profile as per need of the moment, manually... for an electronic check, this pivoting and inverting mech would have to be electrically isolated from the rest of the machine (which experiments continuity throughout the frame so is impossible to know when one part of the frame touches another, as there is already an alternative conductive route for the signal)

A plastic sleeve isolating the fixed side of a hinge and the mobile side that carries the L-profile mech could be a way to later connect said profile to a pin, but it would activate on touching EITHER barrel/nozzle OR platform base, when what is really needed is to assure the profile is correctly positioned between both sides, not just that is contacting either of them

For the moment, can only think about a Manually Display User Reminder to place the L-profile in the correctly orientated position as per the Purge or Compression function that need to be realised...

The same goes for the presence of the mould itself, complicated by the fact that each mould, although metallic, have a wooden base to supplement the height differences between moulds and the full injection space

