Complete Wednesday Evening IDP Changelog

System.ino edits (reasonably major - I think ive kept what worked and improved what didnt)

Turn names

It wouldn't be me coding if I didn't change these again... (sorry).
Most errors were coming from turn instructions that weren't themselves turning, they would be the expect block soon, or right_then_dip or robot_go_wee_woo_after_next.
The initial system didn't have infrastructure for these and would fail to handle them well, often flagging up the failsafe (we will get onto this in a moment). The new turns should be much simpler. I will list them below. Also SPECIAL_DONE exists now as it too is basically another item that uses the new recursive system.

New navigation instructions:

- STRAIGHT_ON
- RIGHT
- LEFT
- **SPECIAL_FROM_THE_LEFT**
- **SPECIAL_FROM THE RIGHT**
- **SPECIAL DONE**
- **TESTING**
- **DROP OFF ANT**
- **EXPECT BLOCK ANT**
- **ROBOT GO WEE WOO ANT**
- **COMPLETED**

ANT stands for After Next Turn
Notice how all the actual turning is literally just three moves, this is how it should have been.
The rest will now edit parameters and call the next move within themselves, this way the junction checkpoint time wont be reset with every new move because to the rest of the world these moves are just a single block of code. (Im quite proud of this solution because it feels proper.) More explained later.

Path adjusted to reflect this.

Iterate_respective_progress

Only called twice in the end after tidying up lots but this code is automated because a couple of errors were likely from me editing progress in special mode etc. Neither progress or special_progress should be directly changed. When this is called with parameter (-1) that means that we were one step further than we thought and should fix it, the junction failsafe main character.

EDITED Get_turn_direction

Implemented iterate respective progress and...
Honestly looking at the old one it was awful, so jumbled, lots of things happened twice and should happen elsewhere...
Removed failsafe checkpoint time updates, as they should be done in system routine once the junction function is finished, as this is the time at which I actually am stating I think that we have finished the junction.
progress++ rubbish removed -- First it is now in the junction function and second it is using my automation to avoid errors, yes it calls one more if statement, the risk of error is worse.
if(direction = STOP) removed from special function, this was another random messy implementation of an ANT function
The code for direction == STOP have been moved to SPECIAL_DONE but didnt work originally from the sounds of it
Serial.println**s** should now be more informative and only printed when of interest.

Edited turn_junction

Pretty major simplifications here.

Starts by iterating progress
Does the case
If it's an ANT itll perform any necessary pre-turn-code before calling turn_junction(....) of the more elemental step, before running any post turn code necessary. This means I can edit the code so much more flexibly, and the odds of any errors are much lower (hopefully 0) the recursion may have some issues, but I doubt it.

Pick up block edited

Actually I havent but I want to make it move_forward_tracking() not move_forward tomorrow.

System_descisions()

There was some pretty temporary code for block picking up where we would go slowly before we picked up the block. This now happens whenever block expected is true. Which feels more universal, and should happen eh same amount. Also the block approach speed is in the config section at the top of the code.

I have also removed has_block as this is based on running the block pickup routine and block drop off routine, so will never have a situation where block_expected is true and !(has_block) is false where it could misinterpret anything as a block. Therefore it is silly to test for both, can be added in if we air grab blocks. Block expected should enable it accurately after turns etc.
Added the last_turn_time_failsafe update to after the turn_junction logic as a separate line in system_navigation, meaning recursive code can work.

IDP_lib edits (small)

Comments

Re-explained is_magnet

Deleted commented code

Deleted hard_stop, was already commented and proven reduntant

go_forward() + state interrupt

Added stop() to go_forward to a new else statement. (Now if it doesn't go, it stops).

LED_blink_async()

Added new functions that attempt to blink the LEDs asynchronously (so we can blink and check sensors with no delay).
The code works via checking if we are in the lower half of every 100ms or upper and sets level accordingly, this should cause a blinking effect when used often enough.
Personally I think a rule of thumb is to minimise delays in the system.

Added new move_speed_parameter to move_forward_tracking()

This is passed into line_track_forward now this too has a parameter in its speed. Nice.
Also the maths for movement delay estimates should be a single function and should be optimised. Not done yet but will be needed for nook block grabbing.

IDP_setup delays reduced and LEDs used as information

Blue led is on for whole setup process.
Red led is on whilst we are waiting for state button to be pressed.
Should be helpful and reduces the artificially increased time of setup, which is already a while because it's a lot of thinking for this board.