

vedderb / rise_sdvp

Finished motor simulator

[Browse files](#)

master

vedderb committed 7 days ago

1 parent 993154b

commit 5539b9133f0e2aeb092a0f1928a93582ad31dc40

Showing 2 changed files with 67 additions and 5 deletions.

Unified Split

2 Embedded/RC_Controller/Makefile

```
285 285 openocd -f stm32-bv_openocd.cfg
286 286
287 287 upload_remote: build/$(PROJECT).bin
288 - ./upload_remote build/$(PROJECT).bin elpgem 10.130.22.7 22 pi_gpio
288 + ./upload_remote build/$(PROJECT).bin elpgem 10.42.0.67 22 pi_gpio
289 289
290 290 RULESPATH = $(CHIBIOS)/os/common/ports/ARMCMx/compilers/GCC
291 291 include $(RULESPATH)/rules.mk
```

70 Embedded/RC_Controller/motor_sim.c

```
5 5 #include "ch.h"
6 6 #include "hal.h"
7 7 #include "bldc_interface.h"
8 8 +#include "utils.h"
9 9
10 10 // Settings
11 11 #define SIMULATION_TIME_MS 10
12 12 +#define MOTOR_KV 520.0
13 13 +#define MOTOR_POLES 4.0
14 14 +#define INPUT_VOLTAGE 39.0
15 15 +#define ERPM_PER_SEC 25000.0
16 16 +#define MAX_CURRENT 80.0
17 17
18 18 // Private variables
19 19 static bool m_is_running;
20 20 +static mc_values m_values;
21 21 +static motor_control_mode m_mode;
22 22 +static float m_mode_value;
23 23
24 24 // Private functions
25 25 static void motor_control_set(motor_control_mode mode, float value);
26 26
27 27 void motor_sim_init(void) {
28 28     m_is_running = false;
29 29     + m_mode = MOTOR_CONTROL_DUTY;
30 30     + m_mode_value = 0.0;
31 31     chThdCreateStatic(sim_thread_wa, sizeof(sim_thread_wa), NORMALPRIO, sim_thread, NULL);
32 32 }
33 33
34 34 systime_t iteration_timer = chVTGetSystemTime();
35 35
36 36 for(;;) {
37 37     + float dt = (float)SIMULATION_TIME_MS / 1000.0;
38 38     +
39 39     if (m_is_running) {
40 40         // TODO!
41 41         const float rpm_max = INPUT_VOLTAGE * MOTOR_KV * (MOTOR_POLES / 2.0);
42 42
43 43         switch (m_mode) {
44 44         case MOTOR_CONTROL_DUTY: {
```

```

66 +         float rpm = m_mode_value * rpm_max;
67 +         utils_step_towards(&m_values.rpm, rpm, ERPM_PER_SEC * dt);
68 +     } break;
69 +
70 +     case MOTOR_CONTROL_CURRENT: {
71 +         utils_step_towards(&m_values.rpm, SIGN(m_mode_value) * rpm_max,
72 +             ERPM_PER_SEC * dt * (fabsf(m_mode_value) / MAX_CURRENT));
73 +     } break;
74 +
75 +     case MOTOR_CONTROL_CURRENT_BRAKE: {
76 +         utils_step_towards(&m_values.rpm, 0.0,
77 +             ERPM_PER_SEC * dt * (fabsf(m_mode_value) / MAX_CURRENT));
78 +     } break;
79 +
80 +     case MOTOR_CONTROL_RPM: {
81 +         utils_step_towards(&m_values.rpm, m_mode_value, ERPM_PER_SEC * dt);
82 +     } break;
83 +
84 +     case MOTOR_CONTROL_POS: {
85 +         // TODO
86 +     } break;
87 +
88 +     default:
89 +         break;
90 +     }
91 +
92 +     // Friction
93 +     m_values.rpm *= powf(0.9, dt);
94 +     utils_step_towards(&m_values.rpm, 0.0, ERPM_PER_SEC * dt * 0.02);
95 +
96 +     // Update values
97 +     m_values.tachometer += m_values.rpm / 60.0 * dt * 6.0;
98 +     m_values.tachometer_abs += fabsf(m_values.rpm) / 60.0 * dt * 6.0;
99 +     m_values.v_in = INPUT_VOLTAGE;
100 +     m_values.duty_now = m_values.rpm / rpm_max;
101 +     m_values.temp_mos = 25.0;
102 +     m_values.temp_motor = 25.0;
103 +     m_values.current_motor = 0.0;
104 +     m_values.current_in = m_values.duty_now * m_values.current_motor;
105 +     m_values.id = 0.0;
106 +     m_values.iq = m_values.current_motor;
107 +     m_values.amp_hours = 0.0;
108 +     m_values.amp_hours_charged = 0.0;
109 +     m_values.watt_hours = 0.0;
110 +     m_values.watt_hours_charged = 0.0;
111 +     m_values.fault_code = FAULT_CODE_NONE;
112 + }
113 +
114 +     iteration_timer = chThdSleepUntilWindowed(iteration_timer,
115 + }
116 +
117 + static void motor_control_set(motor_control_mode mode, float value) {
118 +     (void)mode;
119 +     (void)value;
120 +     m_mode = mode;
121 +     m_mode_value = value;
122 + }
123 +
124 + static void motor_values_requested(void) {
125 +     send_values_to_receiver(&m_values);
126 + }

```

0 comments on commit `5539b91`

