

```

1  //////////////////////////////////////
2  //////////////////////////////////////
3  // Generic list of all parameters that can be changed by marcoq TDSZ-Smart-Ebike controller configurator
4  //////////////////////////////////////
5  //////////////////////////////////////
6  //=====
7  // ENABLES
8  //=====
9  #define ENABLE_VLCD6          1 // enable communication with VLCD6 display
10 #define ENABLE_VLCD5          0 // enable communication with VLCD5 display
11 #define ENABLE_SELECTION3     0 // enable communication with xxxxx display (future implementation)
12 //-----
13 #define ENABLE_LIGHTS_FROM_OEM 1 // enable lights from OEM display (using lights button)
14 #define ENABLE_BACKWARDS_RESISTANCE_OFF 1 // enable backwards resistance off
15 #define ENABLE_STREET_MODE_ON_STARTUP 1 // enable STREET mode on startup
16 //-----
17 #define ENABLE_WALK_ASSIST_FROM_OEM 1 // enable walk assist from OEM display (using walk assist button)
18 #define ENABLE_BRAKE_SENSOR    0 // enable brake sensor
19 #define ENABLE_THROTTLE        0 // enable throttle
20 //-----
21 #define ENABLE_WHEEL_PERIMETER_FROM_OEM 1 // get wheel perimeter from OEM display
22 #define ENABLE_WHEEL_MAX_SPEED_FROM_OEM 1 // get wheel max speed from OEM display
23 //-----
24 #define ENABLE_DISPLAY_WORKING_FLAG 1 // enable working flag when motor run or wheel turn
25 #define ENABLE_DISPLAY_ALWAYS_ON    0 // hold display always on
26 //-----
27 #define STREET_MODE_FEATURE_ENABLED 1 // enable street mode (STREET_CONFIG bit0)
28 #define ENABLE_WALK_ASSIST_OFF_DELAY 0 // enable walk assist off delay
29 #define ENABLE_STREET_POWER_LIMIT   1 // enable street mode power limit (STREET_CONFIG bit2)
30 //-----
31 #define TEMPERATURE_LIMIT_FEATURE_ENABLED 0 // 0 = temperature limit feature disabled, 1 = temperature limit feature
   enabled
32 //-----
33 #define STARTUP_BOOST_WHEN_SPEED_IS_ZERO 1 // enabled on startup when wheel speed is zero
34 #define STARTUP_BOOST_WHEN_CADENCE_IS_ZERO 0 // enable always when cadence was zero
35 //-----
36 // Function Code:
37 #define NO_FUNCTION            0
38 #define BOOST_DISABLED_ON_OEM 1 // E01
39 #define DEFAULT_ENABLED_ON_OEM 2 // E02
40 #define STREET_ENABLED_ON_OEM 3 // E03
41 #define OFFROAD_ENABLED_ON_OEM 4 // E04
42 #define BOOST_ENABLED_ON_OEM 5 // E05
43 //-----
44 // Fault Code:
45 #define NO_FAULT              0
46 #define TEMPERATURE_PROTECTION 6 // E06
47 #define EBIKE_WHEEL_BLOCKED   7 // E07
48 #define OVERVOLTAGE           8 // E08
49 //-----
50

```

```

51 // This file is the firmware configuration for the TSDZ2 motor controller,
52 // to run the 2 different available motors of 36V or 48V motor,
53 // and from 24V battery (7S) up to 52V battery pack (14S).
54
55 //=====
56 // BATTERY
57 //=====
58 // This is the current that motor will draw from the battery
59 // Higher value will give higher torque and the limit of the controller is 16 amps
60 #define ADC_BATTERY_CURRENT_MAX_LIMIT 17.5 // 17.5 amps / 0.625 = 28 (0.625 amps each unit)
61 #define TARGET_MAX_BATTERY_POWER (uint16_t) 625 // 625 watts, 0 is disabled
62 #define BATTERY_MAX_CURRENT_FLOAT 17.0 // 17.0 amps
63 #define BATTERY_CELLS_NUMBER (uint8_t) 10 // 10 cells = 36V
64 #define BATTERY_LOW_VOLTAGE_CUT_OFF_DIV10 29.0 // 36v battery, LVC = 29.0
65 #define BATTERY_PACK_RESISTANCE (uint16_t) 196 // 196 milli ohms, battery pack 36V 10S5P
66
67 // ADC Battery voltage (divisor for cut-off calculation)
68 // 0.344 per ADC_8bits step: 17.9V --> ADC_8bits = 52; 40V --> ADC_8bits = 116; this signal attenuated by the opamp 358
69 #define DIVISOR_FOR_CUTOFF_CALC (uint16_t) 44
70
71 // This values were taken from a discharge graph of Samsung INR18650-25R cells, at almost no current discharge
72 // This graph: https://endless-sphere.com/forums/download/file.php?id=183920&sid=b7fd7180ef87351cabe74a22f1d162d7
73 #define LI_ION_CELL_VOLTS_100 4.25
74 #define LI_ION_CELL_VOLTS_83 3.96
75 #define LI_ION_CELL_VOLTS_50 3.70
76 #define LI_ION_CELL_VOLTS_17 3.44
77 #define LI_ION_CELL_VOLTS_10 3.30
78 #define LI_ION_CELL_VOLTS_0 3.00
79
80 // Possible values: 0, 1, 2, 3, 4, 5, 6
81 // 0 equal to no filtering and no delay, higher values will increase filtering but will also add bigger delay
82 #define SOC_BATTERY_VOLTAGE_FILTER_COEFFICIENT (uint8_t) 4
83 #define SOC_BATTERY_CURRENT_FILTER_COEFFICIENT (uint8_t) 4
84
85 // ADC voltage per ADC step
86 #define SOC_ADC_BATTERY_VOLTAGE_PER_ADC_STEP 0.0866
87
88 // Possible values: 0, 1, 2, 3, 4, 5, 6
89 // 0 equal to no filtering and no delay, higher values will increase filtering but will also add bigger delay
90 #define READ_BATTERY_VOLTAGE_FILTER_COEFFICIENT (uint8_t) 2
91 #define READ_BATTERY_CURRENT_FILTER_COEFFICIENT (uint8_t) 2
92
93 //=====
94 // MOTOR
95 //=====
96 #define MOTOR_TYPE_36V 1 // motor type 36V (CONFIG_1 bit0)
97 #define MOTOR_TYPE_48V 0 // motor type 48V (CONFIG_1 bit0)
98 #define MOTOR_ASSISTANCE_WITHOUT_PEDAL_ROTATION 0 // motor assistance start without pedal rotation (CONFIG_1
bit2)
99 #define MOTOR_MAX_POWER (uint16_t) 250 // 250 watts
100 #define ADC_MOTOR_PHASE_CURRENT_MAX_AMP (uint8_t) 30.0 // 30 amps

```

```

101
102 // Choose some parameters for your motor (if you don't know, just keep the following original values because they should work
    ok)
103 //
104 // This value should be near 0.
105 // You can try to tune with the whell on the air, full throttle and look at battery current: adjust for lower battery current
106 #define MOTOR_ROTOR_OFFSET_ANGLE (uint8_t) 10
107
108 // This value is ERPS speed after which a transition happens from sinewave no interpolation to have
109 // interpolation 60 degrees and must be found experimentally but a value of 25 may be good
110 #define MOTOR_ROTOR_ERPS_START_INTERPOLATION_60_DEGREES (uint8_t) 10
111
112 #define MOTOR_OVER_SPEED_ERPS (uint16_t) 520 // motor max speed, protection max value | 30 points for the
    sinewave at max speed
113 #define MOTOR_OVER_SPEED_ERPS_EXPERIMENTAL (uint16_t) 700 // experimental max motor speed to allow a higher cadence
114
115 //=====
116 // PWM DUTY CYCLE
117 //=====
118 #define PWM_CYCLES_SECOND (uint16_t) 15625 // 1 / 64us(PWM period)
119 #define PWM_DUTY_CYCLE_MIN (uint8_t) 20
120 #define PWM_DUTY_CYCLE_MAX (uint8_t) 254
121
122 // Choose PWM ramp up/down step (higher value will make the motor acceleration slower)
123 //
124 // For a 24V battery, 25 for ramp up seems ok. For an higher voltage battery, this values should be higher
125 #define PWM_DUTY_CYCLE_RAMP_UP_INVERSE_STEP (uint8_t) 38 // 2.5 milliSec
126 #define PWM_DUTY_CYCLE_RAMP_DOWN_INVERSE_STEP (uint8_t) 38 // 2.5 milliSec
127
128 //=====
129 // WHEEL
130 //=====
131 #define WHEEL_PERIMETER (uint16_t) 2083 // 26x2.35 wheel: 2083mm perimeter
132 #define WHEEL_MAX_SPEED (uint8_t) 45 // 45km/h
133 #define OEM_WHEEL_SPEED_FACTOR (uint16_t) 315 // OEM wheel speed factor = 315
134
135 // Wheel speed sensor
136 #define WHEEL_SPEED_SENSOR_MAX_PWM_CYCLE_TICKS (uint16_t) 1166 // something like 100km/h with a 26" wheel
137 #define WHEEL_SPEED_SENSOR_MIN_PWM_CYCLE_TICKS (uint16_t) 65534 // 3.56km/h with a 26" wheel, could be a bigger number,
    // but will make slow detecting wheel stopped
138
139
140 #define WHEEL_SPEED_PI_CONTROLLER_KP_DIVIDEND (uint8_t) 100
141 #define WHEEL_SPEED_PI_CONTROLLER_KP_DIVISOR (uint8_t) 4
142 #define WHEEL_SPEED_PI_CONTROLLER_KI_DIVIDEND (uint8_t) 40
143 #define WHEEL_SPEED_PI_CONTROLLER_KI_DIVISOR (uint8_t) 6
144
145 //=====
146 // PAS
147 //=====
148 // PAS_NUMBER_MAGNETS = 20 was validated on August 2018 by Casainho e jbalat
149 #define PAS_NUMBER_MAGNETS (uint8_t) 20

```

```

150
151 //=====
152 // PEDAL ASSIST
153 //=====
154 #define ASSIST_LEVEL_FACTOR                (uint16_t) 50    // 0.5 = 50%
155 #define ASSIST_LEVEL_FACTOR_X10_1         (uint16_t) 50    // 0.5 = 50%
156 #define ASSIST_LEVEL_FACTOR_X10_2         (uint16_t) 120    // 1.2 = 120%
157 #define ASSIST_LEVEL_FACTOR_X10_3         (uint16_t) 210    // 2.1 = 210%
158 #define ASSIST_LEVEL_FACTOR_X10_4         (uint16_t) 300    // 3.0 = 300%
159
160 //=====
161 // THROTTLE
162 //=====
163 #define ADC_THROTTLE_THRESHOLD            (uint8_t) 10 // value in ADC 8 bits step
164
165 // Possible values: 0, 1, 2, 3, 4, 5, 6
166 // 0 equal to no filtering and no delay, higher values will increase filtering but will also add bigger delay
167 #define THROTTLE_FILTER_COEFFICIENT      (uint8_t) 1
168
169 // Max voltage value for throttle, in ADC 8 bits step
170 // each ADC 8 bits step = (5V / 256) = 0.0195
171 #define ADC_THROTTLE_MIN_VALUE            (uint8_t) 47
172 #define ADC_THROTTLE_MAX_VALUE           (uint8_t) 176
173
174 //=====
175 // TORQUE SENSOR
176 //=====
177 #define ADC_TORQUE_SENSOR_THRESHOLD       (uint8_t) 6 // value in ADC 8 bits step
178
179 // Torque sensor
180 // Torque (force) value found experimentaly
181 // measuring with a cheap digital hook scale, we found that each torque sensor unit is equal to 0.52 Nm
182 // using the scale, was found that each 0.33kg was measured as 1 torque sensor units
183 // Force (Nm) = 1Kg * 9.18 * 0.17 (arm cranks size)
184 #define PEDAL_TORQUE_SENSOR_UNIT          0.52
185
186 // Users did report that pedal human power is about 2x more.
187 // @casainho had the idea to evaluate the torque sensor peak signal (measuring peak signal every pedal rotation)
188 // as being a sinewave and so the average would be:
189 // > [Average value = 0.637 * maximum or peak value, Vpk] (https://www.electronics-tutorials.ws/accircuits/average-voltage.html)
190 // For a quick hack, we can just reduce actual value to 0.637.
191 // 105 * (1/0.637) = 165
192 #define AVERAGE_TORQUE_FACTOR            0.637
193
194 //=====
195 // WALK ASSIST
196 //=====
197 #define WALK_ASSIST_PWM_LEVEL_0           (uint8_t) 10     // 10%
198 #define WALK_ASSIST_PWM_LEVEL_1           (uint8_t) 13     // 13%
199 #define WALK_ASSIST_PWM_LEVEL_2           (uint8_t) 16     // 16%
200 #define WALK_ASSIST_PWM_LEVEL_3           (uint8_t) 19     // 19%

```

```

201 #define WALK_ASSIST_PWM_LEVEL_4                (uint8_t) 22      // 22%
202 #define WALK_ASSIST_PERCENTAGE_CURRENT          (uint8_t) 10      // 10% of max battery current (max = 100%)
203 #define WALK_ASSIST_MAX_RAMP_TIME_DIV10         1.0              // 1.0 seconds
204 #define WALK_ASSIST_OFF_DELAY_PWM_DIV10         2.0              // 2.0 seconds
205
206 // Configure walk assist as throttle with fixed value
207 #define WALK_ASSIST_MIN_VALUE                    (uint8_t) 0
208 #define WALK_ASSIST_MAX_VALUE                    (uint8_t) 100
209
210 //=====
211 // NORMAL MODE
212 //=====
213 #define NORMAL_MODE_SPEED_LIMIT                  (uint8_t) 25     // 25km/h
214 #define NORMAL_MODE_POWER_LIMIT                  (uint16_t) 250    // 250 watts
215
216 //=====
217 // MOTOR POWER BOOST
218 //=====
219 #define STARTUP_MOTOR_POWER_BOOST_LIMIT_MAX_POWER 1               // 0 = disable boost limit max power, 1 = enable boost
limit max power
220 #define STARTUP_MOTOR_BOOST_ASSIST_LEVEL_PERCENT_1 (uint8_t) 100   // 100% = 1450W
221 #define STARTUP_MOTOR_BOOST_ASSIST_LEVEL_PERCENT_2 (uint8_t) 72    // 72% = 1035W
222 #define STARTUP_MOTOR_BOOST_ASSIST_LEVEL_PERCENT_3 (uint8_t) 43    // 43% = 621W
223 #define STARTUP_MOTOR_BOOST_ASSIST_LEVEL_PERCENT_4 (uint8_t) 15    // 15% = 200W
224 #define STARTUP_MOTOR_POWER_BOOST_TIME_DIV10       2.0            // 2.0 seconds, 0 = startup power boost disabled
225 #define STARTUP_MOTOR_POWER_BOOST_FADE_TIME_DIV10  3.5            // 3.5 seconds
226
227 //=====
228 // CRUISE CONTROL
229 //=====
230 #define CRUISE_CONTROL_MIN    (uint8_t) 20
231
232 //=====
233 // TEMPERATURE
234 //=====
235 #define MOTOR_TEMPERATURE_MIN_VALUE_LIMIT           (uint8_t) 75    // 75°C
236 #define MOTOR_TEMPERATURE_MAX_VALUE_LIMIT           (uint8_t) 85    // 85°C
237 #define READ_MOTOR_TEMPERATURE_FILTER_COEFFICIENT   (uint8_t) 4
238
239 //=====
240 // MAGIC BYTE
241 //=====
242 #define MAGIC_BYTE    0xAA
243
244 ////////////////////////////////////////
245 ////////////////////////////////////////
246 // TSDZ2-Smart-Ebike Data Eeprom Map with default values
247 ////////////////////////////////////////
248 ////////////////////////////////////////
249 0x004000 KEY                        170 (0xAA) // KEY = MAGIC_BYTE
250 0x004001 ASSIST LEVEL FACTOR X10     5    (0x05) // 0.5 = 10%
```

251	0x004002	CONFIG_0	0	(0x00)	// bit0 = lights; bit1 = walk assist; bit2 = offroad mode
252	0x004003	BATTERY_MAX_CURRENT	17	(0x11)	// 17 amps
253	0x004004	MOTOR_MAX_POWER_X10	25	(0x19)	// 250 watts
254	0x004005	BATTERY_LOW_VOLTAGE_CUT_OFF_X10_0 (2.9 * 10): (34 + (1 << 8))	34	(0x22)	// Battery low voltage cut-off LSB: 36v battery, LVC = 29.0
255	0x004006	BATTERY_LOW_VOLTAGE_CUT_OFF_X10_1	1	(0x01)	// Battery low voltage cut-off MSB
256	0x004007	WHEEL_PERIMETER_0 (8 << 8))	35	(0x23)	// Wheel perimeter LSB: 26x2.35 wheel: 2083mm perimeter (35 +
257	0x004008	WHEEL_PERIMETER_1	8	(0x08)	// Wheel perimeter MSB
258	0x004009	WHEEL_MAX_SPEED	45	(0x2D)	// 45km/h
259	0x00400A	CONFIG_1	1	(0x01)	// bit0-1 motor_type: 0 = 48V, 1 = 36V; bit2: enable motor
260	0x00400B	OFFROAD_CONFIG	4	(0x04)	// bit0: enable offroad, bit1: enable offroad on startup,
261	0x00400C	OFFROAD_SPEED_LIMIT	25	(0x19)	// 25km/h
262	0x00400D	OFFROAD_POWER_LIMIT_DIV25	10	(0x0A)	// 25 * 10 = 250W
263	0x00400E	BATTERY_CELLS_NUMBER	10	(0x0A)	// 10 cells = 36V
264	0x00400F	BATTERY_PACK_RESISTANCE_0 36V 10S5P	196	(0xC4)	// Battery pack resistance LSB 196 milli ohms, battery pack
265	0x004010	BATTERY_PACK_RESISTANCE_1	0	(0x00)	// Battery pack resistance MSB
266	0x004011	OEM_WHEEL_SPEED_FACTOR_0 (1 << 8))	59	(0x3B)	// Wheel speed factor LSB: OEM wheel speed factor = 315 (59 +
267	0x004012	OEM_WHEEL_SPEED_FACTOR_1	1	(0x01)	// Wheel speed factor MSB
268	0x004013	ASSIST_LEVEL_FACTOR_1	5	(0x05)	// 0.5 = 50%
269	0x004014	ASSIST_LEVEL_FACTOR_2	12	(0x0C)	// 1.2 = 120%
270	0x004015	ASSIST_LEVEL_FACTOR_3	21	(0x15)	// 2.1 = 210%
271	0x004016	ASSIST_LEVEL_FACTOR_4	30	(0x1E)	// 3.0 = 300%
272	0x004017	STARTUP_MOTOR_POWER_BOOST_STATE enable always when cadence was zero	0	(0x00)	// 0 = enabled on startup when wheel speed is zero, 1 =
273	0x004018	STARTUP_MOTOR_POWER_BOOST_FEATURE_ENABLED enabled	0	(0x00)	// 0 = startup power boost disabled, 1 = startup power boost
274	0x004019	STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_1	28	(0x1C)	// 1450W
275	0x00401A	STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_2	20	(0x14)	// 1035W
276	0x00401B	STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_3	12	(0x0C)	// 621W
277	0x00401C	STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_4	4	(0x04)	// 200W
278	0x00401D	STARTUP_MOTOR_POWER_BOOST_TIME	20	(0x14)	// 2.0 seconds, 0 = startup power boost disabled
279	0x00401E	STARTUP_MOTOR_POWER_BOOST_FADE_TIME	35	(0x23)	// 3.5 seconds
280	0x00401F	STARTUP_MOTOR_POWER_BOOST_LIMIT_MAX_POWER max power	1	(0x01)	// 0 = disable boost limit max power, 1 = enable boost limit
281	0x004020	TARGET_MAX_BATTERY_POWER_DIV25	25	(0x19)	// 25 = 625 watts (25 * 25), 0 is disabled
282	0x004021	TEMPERATURE_LIMIT_FEATURE_ENABLED	0	(0x00)	// 0 = disable temperature limit, 1 = enable temperature limit
283	0x004022	MOTOR_TEMPERATURE_MIN_VALUE_LIMIT	75	(0x4B)	// 75°C
284	0x004023	MOTOR_TEMPERATURE_MAX_VALUE_LIMIT	85	(0x55)	// 85°C
285	0x004024	WALK_ASSIST_PERCENTAGE_CURRENT	10	(0x0A)	// 10% of max battery current (max = 100%)
286	0x004025	WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_0	25	(0x19)	// 0...255
287	0x004026	WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_1	33	(0x21)	// 0...255
288	0x004027	WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_2	40	(0x28)	// 0...255
289	0x004028	WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_3	48	(0x30)	// 0...255
290	0x004029	WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_4	56	(0x38)	// 0...255
291	0x00402A	WALK_ASSIST_MAX_RAMP_TIME	10	(0x0A)	// 1.0 seconds
292	0x00402B	WALK_ASSIST_OFF_DELAY_PWM	51	(0x33)	// walk assist off delay PWM (0...255)

```
293 0x00402C WALK_ASSIST_OFF_DELAY_TIME_0
294 0x00402D WALK_ASSIST_OFF_DELAY_TIME_1
295 0x00402E KEY2
296
297
```

```
20 (0x14) // walk assist off delay TIME (LSB)
0  (0x00) // walk assist off delay TIME (MSB)
85 (0x55) // KEY2 = !(MAGIC_BYTE)
```