

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Generic list of all parameters that can be changed by marcoq TDSZ-Smart-Ebike controller configurator
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//=====
// ENABLES
//=====
#define ENABLE_VLCD6                1 // enable communication with VLCD6 display
#define ENABLE_VLCD5                0 // enable communication with VLCD5 display
#define ENABLE_XH18                 0 // enable communication with XH18 display
//-----
#define ENABLE_LIGHTS_FROM_OEM      1 // enable lights from OEM display (using lights button)
#define ENABLE_BACKWARDS_RESISTANCE_OFF 1 // enable backwards resistance off
#define ENABLE_STREET_MODE_ON_STARTUP 1 // enable STREET mode on startup
#define ENABLE_EMTB_MODE_ON_STARTUP 1 // enable eMTB mode on startup
//-----
#define ENABLE_WALK_ASSIST_FROM_OEM 1 // enable walk assist from OEM display (using walk assist button)
#define ENABLE_BRAKE_SENSOR         0 // enable brake sensor
#define ENABLE_THROTTLE             0 // enable throttle
//-----
#define ENABLE_WHEEL_PERIMETER_FROM_OEM 1 // get wheel perimeter from OEM display
#define ENABLE_WHEEL_MAX_SPEED_FROM_OEM 1 // get wheel max speed from OEM display
//-----
#define ENABLE_LAST_BETA_RELEASE    1 // enable last beta code
//-----
#define ENABLE_DISPLAY_WORKING_FLAG 1 // enable working flag when motor run or wheel turn
#define ENABLE_DISPLAY_ALWAYS_ON    0 // hold display always on
//-----
#define STREET_MODE_FEATURE_ENABLED 1 // enable street mode (STREET_CONFIG bit0)
#define ENABLE_WALK_ASSIST_OFF_DELAY 0 // enable walk assist off delay
#define ENABLE_STREET_POWER_LIMIT   1 // enable street mode power limit (STREET_CONFIG bit2)
//-----
#define TEMPERATURE_LIMIT_FEATURE_ENABLED 0 // 0 = temperature limit feature disabled, 1 = temperature limit feature enabled
//-----
#define STARTUP_BOOST_WHEN_SPEED_IS_ZERO 1 // enabled on startup when wheel speed is zero
#define STARTUP_BOOST_WHEN_CADENCE_IS_ZERO 0 // enable always when cadence was zero
//-----
// Function Code for VLCD6 and VLCD5 Display:
#define NO_FUNCTION                0
#define DEFAULT_ENABLED_ON_OEM     2 // E02
#define BOOST_DISABLED_ON_OEM      1 // E02
#define BOOST_ENABLED_ON_OEM       5 // E05
#define STREET_ENABLED_ON_OEM      3 // E03
#define OFFROAD_ENABLED_ON_OEM     4 // E04
#define EMTB_DISABLED_ON_OEM       1 // E01
#define EMTB_ENABLED_ON_OEM        2 // E02

// Function Code for XH18 Display:

```

```

#define NO_FUNCTION 0
#define DEFAULT_ENABLED_ON_OEM 2 // E02
#define BOOST_DISABLED_ON_OEM 2 // E02
#define BOOST_ENABLED_ON_OEM 3 // E03
#define STREET_ENABLED_ON_OEM 2 // E02
#define OFFROAD_ENABLED_ON_OEM 3 // E03
#define EMTB_DISABLED_ON_OEM 2 // E02
#define EMTB_ENABLED_ON_OEM 3 // E03
//-----
// Fault Code for VLCD6 and VLCD5 Display:
#define NO_FAULT 0
#define OVERTEMPERATURE 6 // E06
#define EBIKE_WHEEL_BLOCKED 7 // E07
#define OVERVOLTAGE 8 // E08

// Fault Code for XH18 Display:
#define NO_FAULT 0
#define OVERTEMPERATURE 6 // E06
#define EBIKE_WHEEL_BLOCKED 4 // E04
#define OVERVOLTAGE 8 // E08
//-----

// This file is the firmware configuration for the TSDZ2 motor controller,
// to run the 2 different available motors of 36V or 48V motor,
// and from 24V battery (7S) up to 52V battery pack (14S).

//=====
// BATTERY
//=====
// This is the current that motor will draw from the battery
// Higher value will give higher torque and the limit of the controller is 16 amps
#define ADC_BATTERY_CURRENT_MAX_LIMIT 17.5 // 17.5 amps / 0.625 = 28 (0.625 amps each unit)
#define TARGET_MAX_BATTERY_POWER (uint16_t) 625 // 625 watts, 0 is disabled
#define BATTERY_MAX_CURRENT_FLOAT 17.0 // 17.0 amps
#define BATTERY_CELLS_NUMBER (uint8_t) 10 // 10 cells = 36V
#define BATTERY_LOW_VOLTAGE_CUT_OFF_DIV10 29.0 // 36v battery, LVC = 29.0
#define BATTERY_PACK_RESISTANCE (uint16_t) 196 // 196 milli ohms, battery pack 36V 10S5P

// ADC Battery voltage (divisor for cut-off calculation)
// 0.344 per ADC_8bits step: 17.9V --> ADC_8bits = 52; 40V --> ADC_8bits = 116; this signal attenuated by the opamp 358
#define DIVISOR_FOR_CUTOFF_CALC (uint16_t) 44

// This values were taken from a discharge graph of Samsung INR18650-25R cells, at almost no current discharge
// This graph: https://endless-sphere.com/forums/download/file.php?id=183920&sid=b7fd7180ef87351cabe74a22f1d162d7
#define LI_ION_CELL_VOLTS_100 4.25
#define LI_ION_CELL_VOLTS_83 3.96
#define LI_ION_CELL_VOLTS_50 3.70
#define LI_ION_CELL_VOLTS_17 3.44
#define LI_ION_CELL_VOLTS_10 3.30

```

```

#define LI_ION_CELL_VOLTS_0                3.00

// Possible values: 0, 1, 2, 3, 4, 5, 6
// 0 equal to no filtering and no delay, higher values will increase filtering but will also add bigger delay
#define SOC_BATTERY_VOLTAGE_FILTER_COEFFICIENT    (uint8_t) 4
#define SOC_BATTERY_CURRENT_FILTER_COEFFICIENT    (uint8_t) 4

// ADC voltage per ADC step
#define SOC_ADC_BATTERY_VOLTAGE_PER_ADC_STEP      0.0866

// Possible values: 0, 1, 2, 3, 4, 5, 6
// 0 equal to no filtering and no delay, higher values will increase filtering but will also add bigger delay
#define READ_BATTERY_VOLTAGE_FILTER_COEFFICIENT    (uint8_t) 2
#define READ_BATTERY_CURRENT_FILTER_COEFFICIENT    (uint8_t) 2

//=====
// MOTOR
//=====
#define MOTOR_TYPE_36V                1           // motor type 36V (CONFIG_1 bit0)
#define MOTOR_TYPE_48V                0           // motor type 48V (CONFIG_1 bit0)
#define MOTOR_ASSISTANCE_WITHOUT_PEDAL_ROTATION  0           // motor assistance start without pedal rotation (CONFIG_1 bit2)
#define MOTOR_MAX_POWER                (uint16_t) 250    // 250 watts
#define ADC_MOTOR_PHASE_CURRENT_MAX_AMP    (uint8_t) 30.0 // 30 amps

// Choose some parameters for your motor (if you don't know, just keep the following original values because they should work ok)
//
// This value should be near 0.
// You can try to tune with the whell on the air, full throttle and look at battery current: adjust for lower battery current
#define MOTOR_ROTOR_OFFSET_ANGLE (uint8_t) 10

// This value is ERPS speed after which a transition happens from sinewave no interpolation to have
// interpolation 60 degrees and must be found experimentally but a value of 25 may be good
#define MOTOR_ROTOR_ERPS_START_INTERPOLATION_60_DEGREES (uint8_t) 10

#define MOTOR_OVER_SPEED_ERPS          (uint16_t) 520 // motor max speed, protection max value | 30 points for the sinewave
at max speed
#define MOTOR_OVER_SPEED_ERPS_EXPERIMENTAL (uint16_t) 700 // experimental max motor speed to allow a higher cadence

//=====
// PWM DUTY CYCLE
//=====
#define PWM_CYCLES_SECOND                (uint16_t) 15625 // 1 / 64us(PWM period)
#define PWM_DUTY_CYCLE_MIN                (uint8_t) 20
#define PWM_DUTY_CYCLE_MAX                (uint8_t) 254

#define ADC_BATTERY_CURRENT_RAMP_UP_INVERSE_STEP      5.0
// Choose PWM ramp up/down step (higher value will make the motor acceleration slower)
//
// For a 24V battery, 25 for ramp up seems ok. For an higher voltage battery, this values should be higher

```

```
#define PWM_DUTY_CYCLE_RAMP_UP_INVERSE_STEP      (uint8_t) 20 // 1.3 milliSec
#define PWM_DUTY_CYCLE_RAMP_DOWN_INVERSE_STEP   (uint8_t) 20 // 1.3 milliSec

//=====
// WHEEL
//=====
#define WHEEL_PERIMETER                        (uint16_t) 2083 // 26x2.35 wheel: 2083mm perimeter
#define WHEEL_MAX_SPEED                        (uint8_t) 45 // 45km/h
#define OEM_WHEEL_SPEED_FACTOR                 (uint16_t) 315 // OEM wheel speed factor = 315

// Wheel speed sensor
#define WHEEL_SPEED_SENSOR_MAX_PWM_CYCLE_TICKS (uint16_t) 1166 // something like 100km/h with a 26" wheel
#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

#define WHEEL_SPEED_PI_CONTROLLER_KP_DIVIDEND   (uint8_t) 100
#define WHEEL_SPEED_PI_CONTROLLER_KP_DIVISOR   (uint8_t) 4
#define WHEEL_SPEED_PI_CONTROLLER_KI_DIVIDEND  (uint8_t) 40
#define WHEEL_SPEED_PI_CONTROLLER_KI_DIVISOR   (uint8_t) 6

//=====
// PAS and SOFT START
//=====
// PAS_NUMBER_MAGNETS = 20 was validated on August 2018 by Casainho e jbalat
#define PAS_NUMBER_MAGNETS                     (uint8_t) 20
#define SOFT_START_TIME_RAMP                    30 // 1/10 Second, 0 = Soft Start Ramp disabled
#define K_SOFT_START_FACTOR                     0.990 // K Soft Start Factor
#define THRESHOLD_SOFT_START_PAS_CADENCE        (uint8_t) 10

//=====
// PEDAL ASSIST
//=====
#define MASTER_LEVEL_FACTOR 1.0
#define DEFAULT_ASSIST_LEVEL_FACTOR             (uint16_t) 120 // 1.2 = 120%
#define ASSIST_LEVEL_FACTOR_X10_1               (uint16_t) 50 // 0.5 = 50%
#define ASSIST_LEVEL_FACTOR_X10_2               (uint16_t) 120 // 1.2 = 120%
#define ASSIST_LEVEL_FACTOR_X10_3               (uint16_t) 210 // 2.1 = 210%
#define ASSIST_LEVEL_FACTOR_X10_4               (uint16_t) 300 // 3.0 = 300%

//=====
// THROTTLE
//=====
#define ADC_THROTTLE_THRESHOLD                  (uint8_t) 10 // value in ADC 8 bits step

// Possible values: 0, 1, 2, 3, 4, 5, 6
// 0 equal to no filtering and no delay, higher values will increase filtering but will also add bigger delay
#define THROTTLE_FILTER_COEFFICIENT            (uint8_t) 1

// Max voltage value for throttle, in ADC 8 bits step
```

```

// each ADC 8 bits step = (5V / 256) = 0.0195
#define ADC_THROTTLE_MIN_VALUE          (uint8_t) 47
#define ADC_THROTTLE_MAX_VALUE          (uint8_t) 176

//=====
// TORQUE SENSOR
//=====
#define ADC_TORQUE_SENSOR_THRESHOLD      (uint8_t) 6 // value in ADC 8 bits step

// Torque sensor
// Torque (force) value found experimentaly
// measuring with a cheap digital hook scale, we found that each torque sensor unit is equal to 0.55 Nm
// using the scale, was found that each 0.33kg was measured as 1 torque sensor units
// Force (Nm) = 1Kg * 9.81 * 0.17 (arm cranks size)
#define PEDAL_TORQUE_SENSOR_UNIT         0.55

// Users did report that pedal human power is about 2x more.
// @casainho had the idea to evaluate the torque sensor peak signal (measuring peak signal every pedal rotation)
// as being a sinewave and so the average would be:
// > [Average value = 0.637 × maximum or peak value, Vpk](https://www.electronics-tutorials.ws/ac/circuits/average-voltage.html)
// For a quick hack, we can just reduce actual value to 0.637.
// 105 * (1/0.637) = 165
#define AVERAGE_TORQUE_FACTOR           0.637

//=====
// WALK ASSIST
//=====
#define WALK_ASSIST_PWM_LEVEL_0          (uint8_t) 10 // 10%
#define WALK_ASSIST_PWM_LEVEL_1          (uint8_t) 13 // 13%
#define WALK_ASSIST_PWM_LEVEL_2          (uint8_t) 16 // 16%
#define WALK_ASSIST_PWM_LEVEL_3          (uint8_t) 19 // 19%
#define WALK_ASSIST_PWM_LEVEL_4          (uint8_t) 22 // 22%
#define WALK_ASSIST_PERCENTAGE_CURRENT   (uint8_t) 20 // 20% of max battery current (max = 100%)
#define WALK_ASSIST_MAX_RAMP_TIME_DIV10  1.0 // 1.0 seconds
#define WALK_ASSIST_OFF_DELAY_PWM_DIV10  (uint8_t) 20 // 20%
#define WALK_ASSIST_OFF_DELAY_TIME_DIV10 60.0 // 60.0 seconds

// Configure walk assist as throttle with fixed value
#define WALK_ASSIST_MIN_VALUE             (uint8_t) 0
#define WALK_ASSIST_MAX_VALUE             (uint8_t) 100

//=====
// STREET MODE
//=====
#define STREET_SPEED_LIMIT                (uint8_t) 25 // 25km/h
#define STREET_POWER_LIMIT                (uint16_t) 250 // 250 watts

//=====
// MOTOR POWER BOOST

```



```

start without pedal rotation; bit3: enable temperature limit feature
0x00400B OFFROAD_CONFIG 4 (0x04) // bit0: enable offroad, bit1: enable offroad on startup, bit2:
enable offroad power limit
0x00400C OFFROAD_SPEED_LIMIT 25 (0x19) // 25km/h
0x00400D OFFROAD_POWER_LIMIT_DIV25 10 (0x0A) // 25 * 10 = 250W
0x00400E BATTERY_CELLS_NUMBER 10 (0x0A) // 10 cells = 36V
0x00400F BATTERY_PACK_RESISTANCE_0 196 (0xC4) // Battery pack resistance LSB 196 milli ohms, battery pack 36V 10S5P
0x004010 BATTERY_PACK_RESISTANCE_1 0 (0x00) // Battery pack resistance MSB
0x004011 OEM_WHEEL_SPEED_FACTOR_0 59 (0x3B) // Wheel speed factor LSB: OEM wheel speed factor = 315 (59 + (1 <<
8))
0x004012 OEM_WHEEL_SPEED_FACTOR_1 1 (0x01) // Wheel speed factor MSB
0x004013 ASSIST_LEVEL_FACTOR_1 5 (0x05) // 0.5 = 50%
0x004014 ASSIST_LEVEL_FACTOR_2 12 (0x0C) // 1.2 = 120%
0x004015 ASSIST_LEVEL_FACTOR_3 21 (0x15) // 2.1 = 210%
0x004016 ASSIST_LEVEL_FACTOR_4 30 (0x1E) // 3.0 = 300%
0x004017 STARTUP_MOTOR_POWER_BOOST_STATE 0 (0x00) // 0 = enabled on startup when wheel speed is zero, 1 = enable always
when cadence was zero
0x004018 STARTUP_MOTOR_POWER_BOOST_FEATURE_ENABLED 0 (0x00) // 0 = startup power boost disabled, 1 = startup power boost enabled
0x004019 STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_1 28 (0x1C) // 1450W
0x00401A STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_2 20 (0x14) // 1035W
0x00401B STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_3 12 (0x0C) // 621W
0x00401C STARTUP_MOTOR_POWER_BOOST_ASSIST_LEVEL_4 4 (0x04) // 200W
0x00401D STARTUP_MOTOR_POWER_BOOST_TIME 20 (0x14) // 2.0 seconds, 0 = startup power boost disabled
0x00401E STARTUP_MOTOR_POWER_BOOST_FADE_TIME 35 (0x23) // 3.5 seconds
0x00401F STARTUP_MOTOR_POWER_BOOST_LIMIT_MAX_POWER 1 (0x01) // 0 = disable boost limit max power, 1 = enable boost limit max power
0x004020 TARGET_MAX_BATTERY_POWER_DIV25 25 (0x19) // 25 = 625 watts (25 * 25), 0 is disabled
0x004021 TEMPERATURE_LIMIT_FEATURE_ENABLED 0 (0x00) // 0 = disable temperature limit, 1 = enable temperature limit
0x004022 MOTOR_TEMPERATURE_MIN_VALUE_LIMIT 75 (0x4B) // 75°C
0x004023 MOTOR_TEMPERATURE_MAX_VALUE_LIMIT 85 (0x55) // 85°C
0x004024 WALK_ASSIST_PERCENTAGE_CURRENT 10 (0x0A) // 10% of max battery current (max = 100%)
0x004025 WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_0 25 (0x19) // 0...255
0x004026 WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_1 33 (0x21) // 0...255
0x004027 WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_2 40 (0x28) // 0...255
0x004028 WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_3 48 (0x30) // 0...255
0x004029 WALK_ASSIST_PWM_DUTY_CYCLE_LEVEL_4 56 (0x38) // 0...255
0x00402A WALK_ASSIST_MAX_RAMP_TIME 10 (0x0A) // 1.0 seconds
0x00402B WALK_ASSIST_OFF_DELAY_PWM 51 (0x33) // walk assist off delay PWM (0...255)
0x00402C WALK_ASSIST_OFF_DELAY_TIME_0 20 (0x14) // walk assist off delay TIME (LSB)
0x00402D WALK_ASSIST_OFF_DELAY_TIME_1 0 (0x00) // walk assist off delay TIME (MSB)
0x00402E KEY2 85 (0x55) // KEY2 = !(MAGIC_BYTE)

```