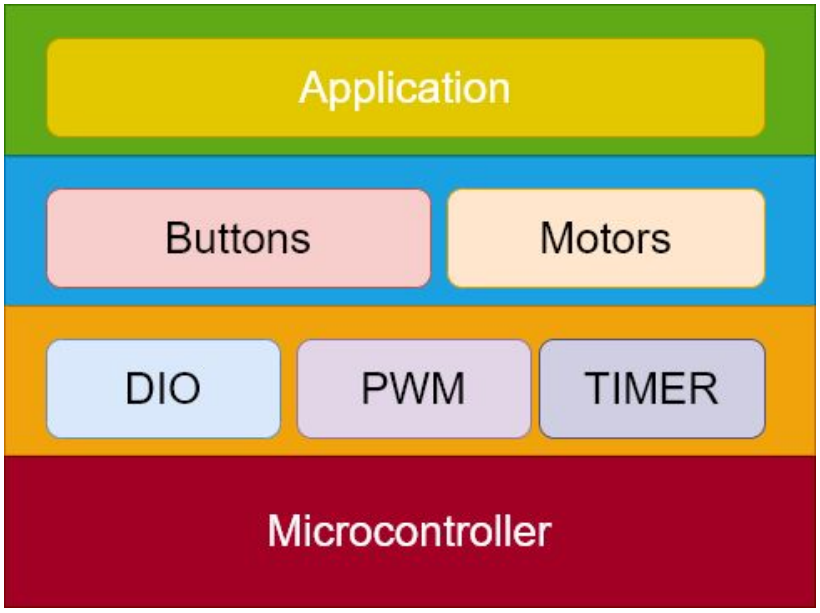
Task 5

Name: Kareem Mohamed Abdelhamid Ragaa Elsawah

High level layered architecture



Module APIs

Application

void App_init(); // Initialize application and call motors and buttons initializers

void App_update(); // Update application every tick

Motors

void motors_init(); // Initialize motors and prepare to take move commands

void motors_update_speed(float speed_1, float speed_2); // Change speeds to 2 motors

Buttons

void buttons_init(); // Initialize 4 buttons and be ready to get their states

EN_button_state* buttons_get_state(); // returns 4 buttons' states (pressed, not pressed)

void buttons_update(); // Update buttons' states

DIO

void DIO_init(ST_DIO_config_t* configurations); // Initialize DIO using configurations

void DIO_read(uint8_t port, EN_pins pin); // Read data from "pin" and "port"

void DIO_write(uint8_t port, EN_pins pin, uint8_t *data); // Write "data" to "pin" and "port"

void DIO_toggle(uint8_t port, EN_pins pin); // Toggle activation of a "pin" and "port"

PWM

void PWM_init(ST_PWM_config_t* configs); // Initialize PWM with configurations

void PWM_start(EN_frequency_t freq, EN_duty_t dutyCycle); // Start PWM signal with frequency and duty cycle

void PWM_stop(); // Stop PWM signal

Timer

void TIMER_init(ST_TIMER_config_t* configs); // Initialize timer with configurations

void TIMER_start(uint64_t ticks); // Start timer ticking to reach ticks

void TIMER_read(uint8_t *value); // Read number of ticks by timer

void TIMER_set(uint8_t value); // Set value in timer

void TIMER_checkStatus(uint8_t *status); // Get timer status