```
#include "wheel.h"

void wheel_init(void){
	RCC->APB2ENR |= RCC_APB2ENR_IOPAEN |  RCC_APB2ENR_IOPBEN; //Enable
GPIOA clock
	GPIOB->CRH |=  GPIO_CRH_MODE8_0 |  GPIO_CRH_MODE8_1;     // Right Wheel
Forward
   GPIOB->CRH &= ~GPIO_CRH_CNF8_0 &~ GPIO_CRH_CNF8_1;
	GPIOB->CRH |=  GPIO_CRH_MODE9_0 |  GPIO_CRH_MODE9_1;     // Right Wheel
Reverse
	GPIOB->CRH &= ~GPIO_CRH_CNF9_0 &~ GPIO_CRH_CNF9_1;
		GPIOA->CRH |=  GPIO_CRH_MODE8_0 |  GPIO_CRH_MODE8_1;     // Left
Wheel Forward
   GPIOA->CRH &= ~GPIO_CRH_CNF8_0 &~ GPIO_CRH_CNF8_1;
	GPIOA->CRH |=  GPIO_CRH_MODE9_0 |  GPIO_CRH_MODE9_1;     // Left Wheel
Reverse
	GPIOA->CRH &= ~GPIO_CRH_CNF9_0 &~ GPIO_CRH_CNF9_1;
}

void delay1(uint32_t delay){while (delay--){}}
void right_wheel_forward_on(void){GPIOB->ODR |= GPIO_ODR_ODR8;delay1(1800);}
void right_wheel_reverse_on(void){GPIOB->ODR |= GPIO_ODR_ODR9;delay1(1800);}
void left_wheel_forward_on(void){GPIOA->ODR |= GPIO_ODR_ODR8;delay1(1800);}
void left_wheel_reverse_on(void){GPIOA->ODR |= GPIO_ODR_ODR9;delay1(1800);}
void right_wheel_forward_off(void){GPIOB->ODR &= (uint32_t)
~GPIO_ODR_ODR8;delay1(1800);}
void right_wheel_reverse_off(void){GPIOB->ODR &= (uint32_t)
~GPIO_ODR_ODR9;delay1(1800);}
void left_wheel_forward_off(void){GPIOA->ODR &= (uint32_t)
~GPIO_ODR_ODR8;delay1(1800);}
void left_wheel_reverse_off(void){GPIOA->ODR &= (uint32_t)
~GPIO_ODR_ODR9;delay1(1800);}

void stop(void){
		// Turn off all tires
		left_wheel_forward_off();
		left_wheel_reverse_off();
		right_wheel_forward_off();
		right_wheel_reverse_off();
}

void turn_left(void){

		left_wheel_reverse_on();
		delay(1400000);
		left_wheel_reverse_off();
		delay(1050000);
		right_wheel_forward_on();
		left_wheel_forward_on();

}

void turn_right(void){
```

```
                right_wheel_reverse_on();
                delay(1800000);
                right_wheel_reverse_off();
                delay(1050000);
                right_wheel_forward_on();
                left_wheel_forward_on();

}
```