```
#include <Arduino.h>
#include "../vars/constants.h"

#define LED_BLINK_DELAY 100

class Led
{
    private:
    TaskHandle_t blinkLEDsTask = NULL;

    static void BlinkLEDsTask(void * parameter)
    {
        bool topLedFlag = false;
        while(!false)
        {
            led_1_on();
            vTaskDelay(LED_BLINK_DELAY / portTICK_PERIOD_MS);
            if(topLedFlag)
                led_top_on();
            led_1_off();
            vTaskDelay(LED_BLINK_DELAY / portTICK_PERIOD_MS);
            led_1_on();
            vTaskDelay(LED_BLINK_DELAY / portTICK_PERIOD_MS);
            if(topLedFlag)
                led_top_off();
            led_1_off();

            vTaskDelay((LED_BLINK_DELAY * 2) / portTICK_PERIOD_MS);

            led_2_on();
            vTaskDelay(LED_BLINK_DELAY / portTICK_PERIOD_MS);
            if(!topLedFlag)
                led_top_on();
            led_2_off();
            vTaskDelay(LED_BLINK_DELAY / portTICK_PERIOD_MS);
            led_2_on();
            vTaskDelay(LED_BLINK_DELAY / portTICK_PERIOD_MS);
            if(!topLedFlag)
                led_top_off();
            led_2_off();

            vTaskDelay((LED_BLINK_DELAY * 2) / portTICK_PERIOD_MS);

            topLedFlag = !topLedFlag;
        }
    }

    public:
    bool isBlinking()
    {
        if(blinkLEDsTask == NULL)
            return false;

```

```
54          auto taskState = eTaskGetState(blinkLEDsTask);
55
56          return taskState != eDeleted && taskState != eReady &&
              taskState != eSuspended;
57      }
58
59      static void led_top_on()
60      {
61          digitalWrite(constants::pins::led::Top, HIGH);
62      }
63      static void led_1_on()
64      {
65          digitalWrite(constants::pins::led::Led1, HIGH);
66      }
67      static void led_2_on()
68      {
69          digitalWrite(constants::pins::led::Led2, HIGH);
70      }
71      static void led_top_off()
72      {
73          digitalWrite(constants::pins::led::Top, LOW);
74      }
75      static void led_1_off()
76      {
77          digitalWrite(constants::pins::led::Led1, LOW);
78      }
79      static void led_2_off()
80      {
81          digitalWrite(constants::pins::led::Led2, LOW);
82      }
83
84      Led()
85      {
86          pinMode(constants::pins::led::Top, OUTPUT);
87          pinMode(constants::pins::led::Led1, OUTPUT);
88          pinMode(constants::pins::led::Led2, OUTPUT);
89          led_top_off();
90          led_1_off();
91          led_2_off();
92      }
93
94      void StartBlink()
95      {
96          if(isBlinking())
97              return;
98
99          xTaskCreatePinnedToCore(
100             BlinkLEDsTask,
101             "BlinkLEDsTask",
102             1000,
103             NULL,
104             0,
105             &blinkLEDsTask,
```

```
106             0);
107     }
108
109     void StopBlink()
110     {
111         if(isBlinking())
112         {
113             vTaskDelete(blinkLEDsTask);
114         }
115         led_top_off();
116         led_1_off();
117         led_2_off();
118     }
119
120     void Toggle()
121     {
122         if(isBlinking())
123             StopBlink();
124         else
125             StartBlink();
126     }
127 };
```