

```

1  /* USER CODE BEGIN Header */
2  /**
3   * *****
4   * @file      : main.c
5   * @brief     : Main program body
6   * *****
7   * @attention
8   *
9   * <h2><center>&copy; Copyright (c) 2021 STMicroelectronics.
10  * All rights reserved.</center></h2>
11  *
12  * This software component is licensed by ST under BSD 3-Clause license,
13  * the "License"; You may not use this file except in compliance with the
14  * License. You may obtain a copy of the License at:
15  *      opensource.org/licenses/BSD-3-Clause
16  *
17  * *****
18  */
19  /* USER CODE END Header */
20  /* Includes -----*/
21  #include "main.h"
22  #include "adc.h"
23  #include "dma.h"
24  #include "tim.h"
25  #include "gpio.h"
26
27  /* Private includes -----*/
28  /* USER CODE BEGIN Includes */
29  #include "led.h"
30  #include "interrupt.h"
31  #include "stdio.h"
32
33  /* USER CODE END Includes */
34
35  /* Private typedef -----*/
36  /* USER CODE BEGIN PTD */
37  extern struct keys key[4];
38
39  /* USER CODE END PTD */
40
41  /* Private define -----*/
42  /* USER CODE BEGIN PD */
43  #define PA7_FREQ 1000000
44  /* USER CODE END PD */
45
46  /* Private macro -----*/
47  /* USER CODE BEGIN PM */
48
49  /* USER CODE END PM */
50
51  /* Private variables -----*/
52
53  /* USER CODE BEGIN PV */
54  char lcd_array[50];
55  char lcd_view = 0; //1:费用明细
56  uint16 adc1_array[1];
57  uint16 adc2_array[1];
58  float fees_value;
59  float kilo_value;
60  uint start_value = 12;
61  float renew_value = 2.5;
62  uchar fees_flag = 0; //1:启动状态
63  uint pa7_autoreload, pa7_compare, pa7_freq;
64  uint car_speed;
65  float car_time;
66  float R37_vol, R38_vol;
67
68
69
70
71  /* USER CODE END PV */

```

```

72
73 /* Private function prototypes -----*/
74 void SystemClock_Config(void);
75 /* USER CODE BEGIN PFP */
76
77 /* USER CODE END PFP */
78
79 /* Private user code -----*/
80 /* USER CODE BEGIN 0 */
81 void lcd_proc(void);
82 void key_proc(void);
83 void pwm_proc(void);
84 void led_proc(void);
85 void adc_proc(void);
86
87 /* USER CODE END 0 */
88
89 /**
90  * @brief The application entry point.
91  * @retval int
92  */
93 int main(void)
94 {
95     /* USER CODE BEGIN 1 */
96
97     /* USER CODE END 1 */
98
99     /* MCU Configuration-----*/
100
101     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
102     HAL_Init();
103
104     /* USER CODE BEGIN Init */
105
106     /* USER CODE END Init */
107
108     /* Configure the system clock */
109     SystemClock_Config();
110
111     /* USER CODE BEGIN SysInit */
112
113     /* USER CODE END SysInit */
114
115     /* Initialize all configured peripherals */
116     MX_GPIO_Init();
117     MX_DMA_Init();
118     MX_TIM2_Init();
119     MX_TIM6_Init();
120     MX_TIM17_Init();
121     MX_ADC2_Init();
122     MX_ADC1_Init();
123     /* USER CODE BEGIN 2 */
124
125     LCD_Init();
126     /* USER CODE END 2 */
127
128     /* Infinite loop */
129     /* USER CODE BEGIN WHILE */
130
131     LCD_Clear(Black);
132     LCD_SetBackColor(Black);
133     LCD_SetTextColor(White);
134
135     HAL_TIM_Base_Start_IT(&htim6);
136
137     HAL_TIM_PWM_Start(&htim17, TIM_CHANNEL_1);
138
139     HAL_ADC_Start_DMA(&hadc1, (uint32_t *)adc1_array, 1);
140     HAL_ADC_Start_DMA(&hadc2, (uint32_t *)adc2_array, 1);
141
142     while (1)

```

```

143 {
144     /* USER CODE END WHILE */
145
146     /* USER CODE BEGIN 3 */
147     lcd_proc();
148     key_proc();
149     pwm_proc();
150     adc_proc();
151 }
152
153 /* USER CODE END 3 */
154 }
155
156 /**
157  * @brief System Clock Configuration
158  * @retval None
159  */
160 void SystemClock_Config(void)
161 {
162     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
163     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
164
165     /** Configure the main internal regulator output voltage
166     */
167     HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR_VOLTAGE_SCALE1);
168
169     /** Initializes the RCC Oscillators according to the specified parameters
170     * in the RCC_OscInitTypeDef structure.
171     */
172     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
173     RCC_OscInitStruct.HSEState = RCC_HSE_ON;
174     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
175     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
176     RCC_OscInitStruct.PLL.PLLM = RCC_PLLM_DIV3;
177     RCC_OscInitStruct.PLL.PLLN = 20;
178     RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
179     RCC_OscInitStruct.PLL.PLLQ = RCC_PLLQ_DIV2;
180     RCC_OscInitStruct.PLL.PLLR = RCC_PLLR_DIV2;
181     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
182     {
183         Error_Handler();
184     }
185
186     /** Initializes the CPU, AHB and APB buses clocks
187     */
188     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYCLK
189                                | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2;
190     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
191     RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
192     RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
193     RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
194
195     if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
196     {
197         Error_Handler();
198     }
199 }
200
201 /* USER CODE BEGIN 4 */
202 void adc_proc()
203 {
204     R37_vol = adc2_array[0]*3.3/4096;
205     R38_vol = adc1_array[0]*3.3/4096;
206
207     if(R37_vol<1)
208         car_speed = 40;
209     else if(R37_vol<3)
210         car_speed = 15*R37_vol + 25;
211     else car_speed = 70;
212
213     car_time = R38_vol;

```

```

214 kilo_value = car_time*car_speed;//里程
215
216 if(fees_flag == 1)//启动计价状态下
217 {
218     if(kilo_value<3)
219         fees_value = start_value;
220     else fees_value = start_value + (kilo_value-3)*renew_value;
221 }
222
223
224
225 }
226 void led_proc()
227 {
228
229 }
230 void pwm_proc()
231 {
232     if(fees_flag == 0)
233         pa7_freq = 1000;
234     else pa7_freq = 2000;
235
236     pa7_autoreload = PA7_FREQ/pa7_freq;
237     pa7_compare = 50*pa7_autoreload/100;
238
239     __HAL_TIM_SetAutoreload(&htim17, pa7_autoreload);
240     __HAL_TIM_SetCompare(&htim17, TIM_CHANNEL_1, pa7_compare);
241
242 }
243 void key_proc()
244 {
245     for(int i=0;i<4;i++)
246     {
247         if(key[i].short_flag == 1)
248             LCD_Clear(Black);
249     }
250
251     if(key[3].short_flag == 1)
252     {
253         key[3].short_flag = 0;
254         lcd_view = !lcd_view;
255     }
256
257     if(key[0].short_flag == 1)
258     {
259         key[0].short_flag = 0;
260         if(lcd_view == 0)//计价界面下
261         {
262             fees_flag = !fees_flag;
263         }
264     }
265
266
267     if(key[1].short_flag == 1)
268     {
269         key[1].short_flag = 0;
270         if(fees_flag == 0)//停止计价状态
271             fees_value = 0;
272     }
273
274     if(key[2].short_flag == 1)
275     {
276         key[2].short_flag = 0;
277
278         if(lcd_view == 1)
279         {
280             renew_value+=0.5;
281             if((uint)(renew_value*10) == 40)
282             {
283                 renew_value = 1.5;
284             }

```

```

285     }
286
287 }
288
289 }
290
291 void lcd_proc ()
292 {
293     if (lcd_view == 0) //计价界面
294     {
295         sprintf (lcd_array, "      Fees");
296         LCD_DisplayStringLine (Line2, (unsigned char *) lcd_array);
297         sprintf (lcd_array, "$:%-6.1f", fees_value);
298         LCD_DisplayStringLine (Line3, (unsigned char *) lcd_array);
299     }
300     if (lcd_view == 1) //费用明细界面
301     {
302         sprintf (lcd_array, "      Detail");
303         LCD_DisplayStringLine (Line2, (unsigned char *) lcd_array);
304         sprintf (lcd_array, "      Kilo:%-6.1f", kilo_value);
305         LCD_DisplayStringLine (Line3, (unsigned char *) lcd_array);
306         sprintf (lcd_array, "      Start:%-6d", start_value);
307         LCD_DisplayStringLine (Line4, (unsigned char *) lcd_array);
308         sprintf (lcd_array, "      Renew:%-6.1f", renew_value);
309         LCD_DisplayStringLine (Line5, (unsigned char *) lcd_array);
310     }
311 }
312
313 /* USER CODE END 4 */
314
315 /**
316  * @brief This function is executed in case of error occurrence.
317  * @retval None
318  */
319 void Error_Handler(void)
320 {
321     /* USER CODE BEGIN Error_Handler_Debug */
322     /* User can add his own implementation to report the HAL error return state */
323
324     /* USER CODE END Error_Handler_Debug */
325 }
326
327 #ifdef USE_FULL_ASSERT
328 /**
329  * @brief Reports the name of the source file and the source line number
330  * where the assert_param error has occurred.
331  * @param file: pointer to the source file name
332  * @param line: assert param error line source number
333  * @retval None
334  */
335 void assert_failed(uint8_t *file, uint32_t line)
336 {
337     /* USER CODE BEGIN 6 */
338     /* User can add his own implementation to report the file name and line number,
339        tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
340     /* USER CODE END 6 */
341 }
342
343 #endif /* USE_FULL_ASSERT */
344

```