

Использование параметрически заданных функций для реализации периодических процессов

Ярмолинский А.М.

10 июня 2024 г.

1 Мигание четырьмя светодиодами по очереди

Предположим, у нас есть 4 светодиода, которые мы хотим зажигать по очереди.

1.1 Наивный подход

Стандартный подход к решению такой задачи может выглядеть примерно так:

```
1 int LED[4] = {...};
2 void setup()
3 {
4     pinMode(LED[0], OUTPUT);
5     pinMode(LED[1], OUTPUT);
6     pinMode(LED[2], OUTPUT);
7     pinMode(LED[3], OUTPUT);
8 }
9
10 void loop()
```

```

11 {
12     for(int i = 0; i < 4; i++)
13     {
14         digitalWrite(LED[(i+1) % 4], HIGH);
15         digitalWrite(LED[i], LOW);
16         delay(1000);
17     }
18 }

```

Листинг 1: Пример мигания четырьмя светодиодами

Такое решение работает очень хорошо, до тех пор пока все что нам нужно делать - дискретно мигать четырьмя светодиодами по кругу. А теперь мы хотим добавить кнопку, по нажатию на которую направление движения огонька изменяется, при этом не теряя его положения. Как мы можем это сделать?

Для решения подобных задач может пригодится метод с использованием параметрически заданных функций.

1.2 Немного теории

Параметрически заданная функция - функция, заданная в виде зависимости

$$y = f(t),$$

где t - некоторый параметр, который может меняться в произвольных заранее заданных пределах.

1.3 И зачем нам это?

Чем же это может помочь в задаче мигания светодиодами?

Для начала необходимо понять, а что означает мигание светодиодами по кругу. Заметим, что это процесс периодический. А значит мы можем рассматривать только время в рамках одного периода.

Посмотрим на то, как выглядит яркость одного из светодиодов в зависимости от времени в рамках одного периода (рис. 1):

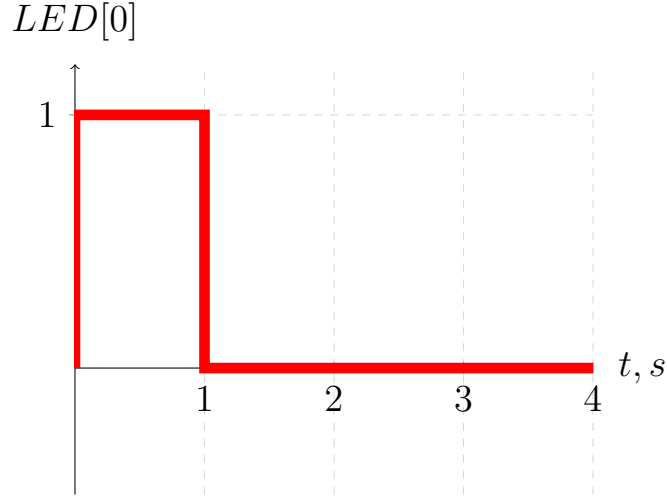


Рис. 1: яркость одного из светодиодов в рамках периода в зависимости от времени

Проанализировав график мы можем заметить, что яркость зависит от времени следующей зависимостью (при $t \in [0, 4)s$):

$$L_0(t) = \begin{cases} 1, & t \leq 1 \\ 0, & t > 1 \end{cases}.$$

1.4 Пример решения

Для решения нашей задачи можно эту функцию реализовать в микроконтроллере напрямую, однако я бы сделал одно упрощение, которое в конкретно данном случае с моей точки зрения имеет смысл рассмотреть. Мы зададим параметр t как целое число счетчик ($t \in \{0, 1, 2, 3\}$).

Функции яркости светодиодов можно в таком случае задать как:

$$L_i(t) = \begin{cases} 1, & t = i \\ 0, & t \neq i \end{cases}.$$

Код программы, реализующей данный подход представлен ниже:

```
1 int LED[4] = {...};
2 void setup()
3 {
4     pinMode(LED[0], OUTPUT);
5     pinMode(LED[1], OUTPUT);
6     pinMode(LED[2], OUTPUT);
7     pinMode(LED[3], OUTPUT);
8 }
9
10 bool Li(int t, int i)
11 {
12     return t == 0;
13 }
14
15 int t = 0;
16
17 void loop()
18 {
19     digitalWrite(LED[0], Li(t, 0));
20     digitalWrite(LED[1], Li(t, 1));
21     digitalWrite(LED[2], Li(t, 2));
22     digitalWrite(LED[3], Li(t, 3));
23
24     t++;
25     t &= 0b11;
26
27     delay(1000);
28 }
```

Листинг 2: Пример мигания четырьмя светодиодами с использованием параметрических функций

1.5 Мигаем в другую сторону!

Теперь задача менять направление зажигания светодиодов сводится к просто изменению направления изменения нашего параметра t . Чтобы начать мигать в другую сторону нам достаточно начать его декрементировать, вместо инкрементирования.

```
1 int LED[4] = {...};
2 int BUTTON = ...;
```

```

3 void setup()
4 {
5     pinMode(LED[0], OUTPUT);
6     pinMode(LED[1], OUTPUT);
7     pinMode(LED[2], OUTPUT);
8     pinMode(LED[3], OUTPUT);
9
10    pinMode(BUTTON, INPUT_PULLUP);
11 }
12
13 bool Li(int t, int i)
14 {
15     return t == 0;
16 }
17
18 int t = 0;
19
20 void loop()
21 {
22     bool button = !digitalRead(BUTTON);
23
24     digitalWrite(LED[0], Li(t, 0));
25     digitalWrite(LED[1], Li(t, 1));
26     digitalWrite(LED[2], Li(t, 2));
27     digitalWrite(LED[3], Li(t, 3));
28
29     t += button ? -1 : 1;
30     t &= 0b11;
31
32     delay(1000);
33 }

```

Листинг 3: Пример мигания четырьмя светодиодами с использованием параметрических функций и кнопкой

2 Мигаем плавно

А теперь предположим, что мы хотим мигать этими светодиодами плавно.

2.1 Наивный подход

Наивное решение этой задачи не сильно отличается от решения предыдущей:

```
1 int LED[4] = {...};
2 void setup()
3 {
4     pinMode(LED[0], OUTPUT);
5     pinMode(LED[1], OUTPUT);
6     pinMode(LED[2], OUTPUT);
7     pinMode(LED[3], OUTPUT);
8 }
9
10 void loop()
11 {
12     for(int i = 0; i < 4; i++)
13     {
14         for(int j = 0; j < 255; j++)
15         {
16             analogWrite(LED[(i+1) % 4], j);
17             analogWrite(LED[i], 255 - j);
18             delay(4);
19         }
20     }
21 }
```

Листинг 4: Наивное плавное мигание

Если в предыдущем случае добавить кнопку в наивное решение еще можно как-то умудриться, то здесь я уже даже примерно не могу придумать как это можно сделать не меняя фундаментально структуру кода (так, чтобы мы могли изменить направление движения в ЛЮБОЙ момент с сохранением текущего состояния)

2.2 Анализируем циклограмму

И, чтобы не терять время, сразу перейдем к анализу циклограммы нулевого светодиода в рамках одного периода (рис. 2):

Функционально эту зависимость можно представить в следующем виде:

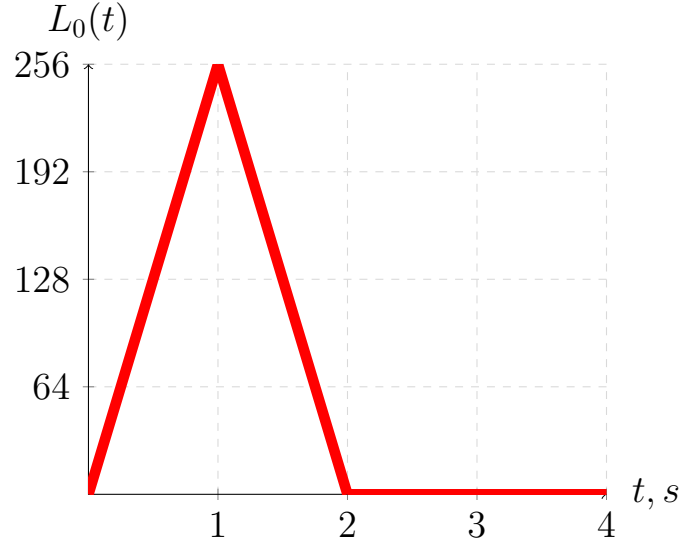


Рис. 2: Циклограмма яркости нулевого светодиода

$$L_0(t) = \begin{cases} At/T, & t \in [0, T/4) \\ A(T - t/T), & t \in [T/4, T/2) , \\ 0, & t \in [T/2, T) \end{cases}$$

где A - амплитуда выходного сигнала (его максимальное значение; 256 в нашем случае), T - период сигнала.

2.3 А как же другие светодиоды?

Получить функцию яркости для других светодиодов можно с помощью манипуляции переменной-параметром. Например функцию для светодиода 1 можно получить следующим образом:

$$L_1(t) = L_0((t - T/4) \bmod T)$$

Аналогично можно получить функции для оставшихся светодиодов. В общем случае можно записать:

$$L_i(t) = L_0((t - iT/4) \bmod T).$$

И выглядеть эти функции будут следующим образом (рис. 3):

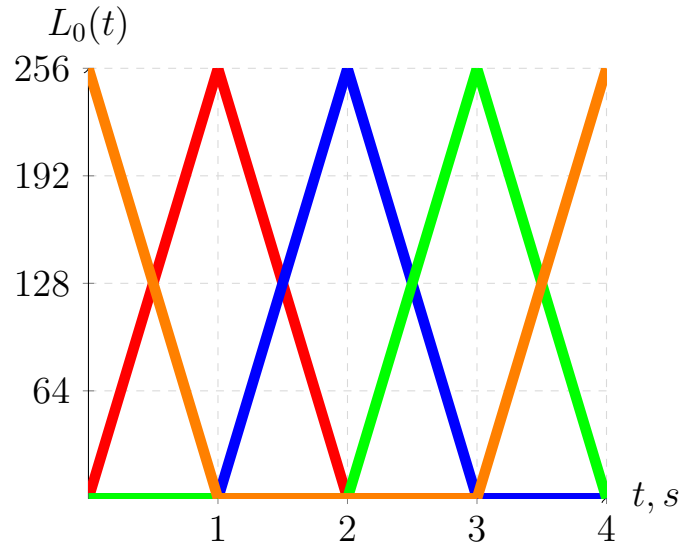


Рис. 3: Циклограмма яркости светодиодов

2.4 Пример решения

Здесь реализуем решение сразу с кнопкой, чтобы не тратить ничье время :).

```

1 int LED[4] = {...};
2 int BUTTON = ...;
3 void setup()
4 {
5     pinMode(LED[0], OUTPUT);
6     pinMode(LED[1], OUTPUT);
7     pinMode(LED[2], OUTPUT);
8     pinMode(LED[3], OUTPUT);
9
10    pinMode(BUTTON, INPUT_PULLUP);
11 }
12

```



```

13 int T = 1024;
14 int t = 0;
15 int A = 256;
16
17 int L0(int t)
18 {
19     t %= T;
20     if(t < T/4)
21     {
22         return A*t/T;
23     }
24     else if(t < T/2)
25     {
26         return A*T - A*t/T;
27     }
28     else
29     {
30         return 0;
31     }
32 }
33
34 int Li(int t, int i)
35 {
36     return L0((t + (4-i)*T/4) % T);
37 }
38
39 void loop()
40 {
41     bool button = !digitalRead(BUTTON);
42
43     analogWrite(LED[0], Li(t, 0));
44     analogWrite(LED[1], Li(t, 1));
45     analogWrite(LED[2], Li(t, 2));
46     analogWrite(LED[3], Li(t, 3));
47
48     t += button ? -1 : 1;
49     t += (t < 0)*T;
50     t %= T;
51
52     delay(1);
53 }

```

Листинг 5: Мигаем плавно вместе с кнопкой

3 Заключение

Данный подход к решению задачи позволяет очень легко изменять вид кривых яркости, да и не только яркости. Такой подход можно использовать в генерации света, звука, периодических движений электроприводом и еще много где.

В качестве домашнего задания читателю предлагается дополнить программу для реализации следующей циклограммы (рис. 4):

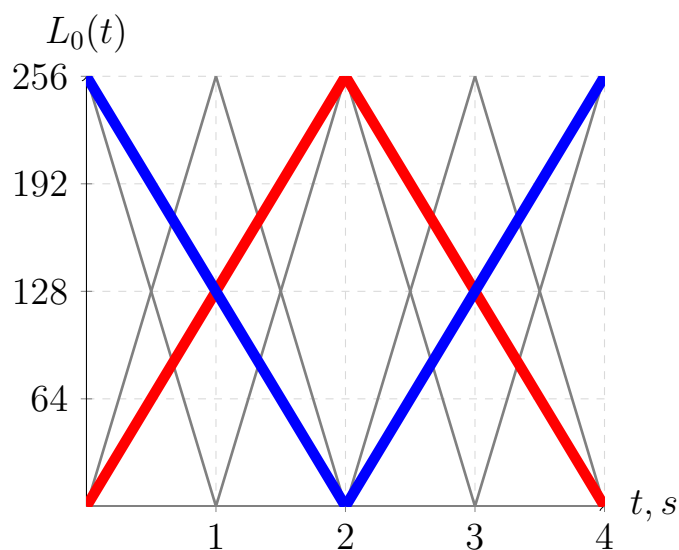


Рис. 4: Домашнее задание (добавлены еще два светодиода; всего должно быть 6 светодиодов)