

Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Факультет Електроніки
Кафедра мікроелектроніки

Домашня контрольна робота
з дисципліни: «Мікропроцесори та мікроконтролери»
Секундомір

Виконав:

Студент 4-го курсу

(підпис)

Мнацаканов А.С.

Перевірив:

(підпис)

Татарчук Д.Д.

1. Мета роботи

Розробити секундомір

2. Розробка схеми

Все що нам потрібно для секундоміра - це те, куди ми могли б виводити інформацію. Найкращим рішенням було використати символьний дисплей 1602 з інтерфесом I2C. Взагалі нам потрібно виводити лише цифри, для цього підійшов би і чотирьохрозрядний семисегментний індикатор а також здвиговий регістр 74НС595, але перший варіант найпростіший. Формуємо схему:

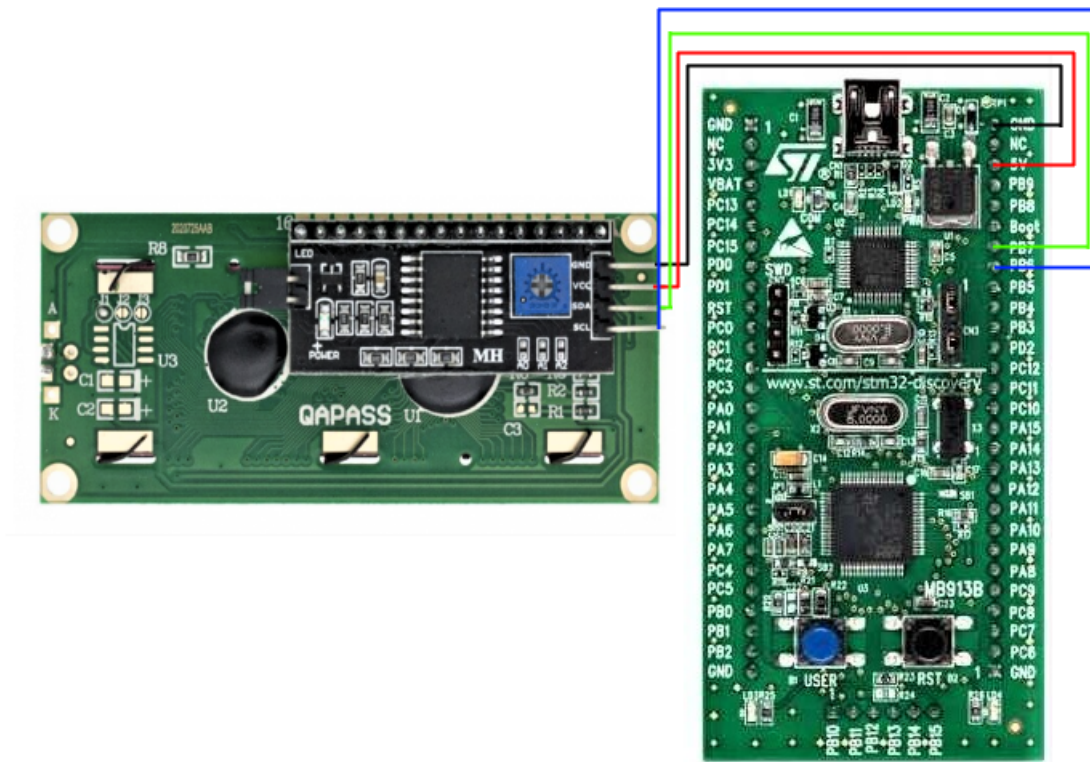


Рис. 1: Сформована схема для секундоміра.

3. Створення проекту

Піни PB7 та PB6 відповідають за лінії SDA та SCL. У розділі System Core: RCC обираємо пункт High Speed Clock (HSE) *CrystalCeramicResonator*. Також налаштуємо кнопку як цифровий вхід. Будемо зчитувати її положення - якщо на неї натиснути - висвітиться результат і відлік почнеться спочатку.

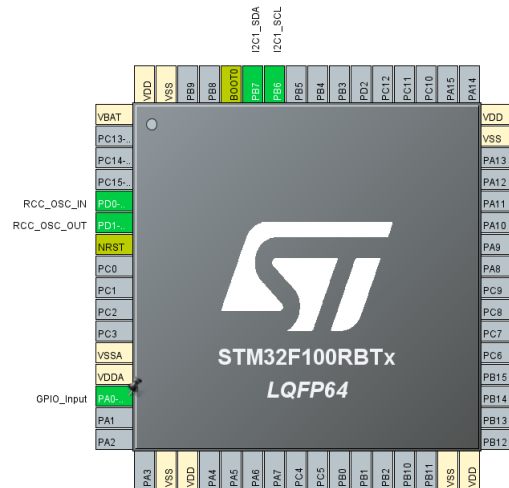


Рис. 2: Налаштування ніжок мікроконтролера.

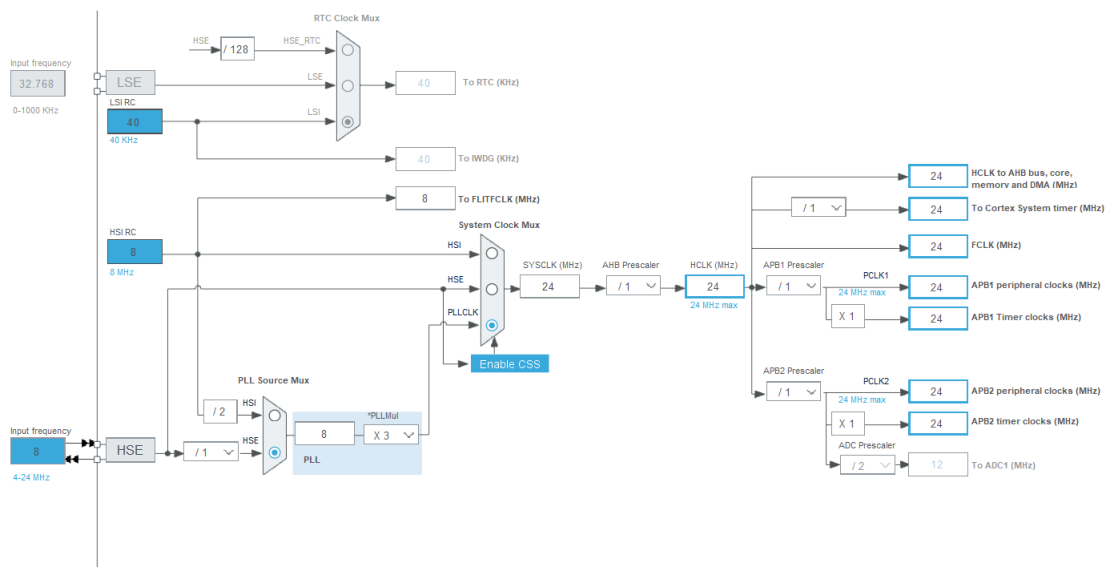


Рис. 3: Налаштування тактування.

5. Програмна частина

Розпишемо яка частина коду за що відповідає.

```
1
2 /* USER CODE END Header */
3 /* Includes ----- */
4 #include <main.h>
5 #include <stdio.h>
6 #include <lcd1602_i2c_lib.h>
7 #include <math.h>
8
9 extern char lcd1602_tx_buffer[80]; //создаём глобальный буфер данных
10
11 int k; //переменная-счётчик
12 int min;
13 int sec;
14
15 /* две вспомогательные переменные для обработчика кнопки */
16 int but_prev = 0 ;
17 int but_cur = 0 ;
18
19
20
21
22 /* Private includes ----- */
```

```
1 /* USER CODE BEGIN WHILE */
2 while (1)
3 {
4
5     but_cur = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ; //счит. тек-е состояние кнопки
6
7     if ( ( but_prev == 0 ) && ( but_cur != 0 ) ) //если нажали
8     {
9         lcd1602_Clean(); //очищаем дисплей
10
11         if(k>=60) //если счётчик больше минуты - выполняем преобразования в мин. и сек.
12         {
13             min = k/60;
14             sec = k % 60;
15
16             lcd1602_SetCursor(0,1);
17             sprintf(lcd1602_tx_buffer, "%d_m", min);
```

```

18         lcd1602_Print_text(lcd1602_tx_buffer);
19
20         lcd1602_SetCursor(5,1);
21         sprintf(lcd1602_tx_buffer, "%d_s", sec);
22         lcd1602_Print_text(lcd1602_tx_buffer);
23
24     }
25
26     else //если нет, то просто выводим результат в секундах
27     {
28         sec = k % 60;
29
30         lcd1602_SetCursor(0,1);
31         sprintf(lcd1602_tx_buffer, "0_m", min);
32         lcd1602_Print_text(lcd1602_tx_buffer);
33
34         lcd1602_SetCursor(5,1);
35         sprintf(lcd1602_tx_buffer, "%d_s", sec);
36         lcd1602_Print_text(lcd1602_tx_buffer);
37     }
38
39     k=0;
40     HAL_Delay(3000);
41     lcd1602_Clean();
42 }
43
44 else //если кнопку не трогаем, счётчик растёт и высвечивается на дисплее
45 {
46     lcd1602_SetCursor(0,0);
47     sprintf(lcd1602_tx_buffer, "Time: %d_s", k);
48     lcd1602_Print_text(lcd1602_tx_buffer);
49     HAL_Delay(1000);
50
51     k++;
52 }
53
54 but_prev = but_cur ; //возвращаем кнопку в неактивное состояние
55
56
57 /* USER CODE END WHILE */
58
59 /* USER CODE BEGIN 3 */
60 }
61 /* USER CODE END 3 */

```