Руководство по работе с дисплеем P10

Введение.

Руководство описывает устройство, работу и программирование дисплея HP10 (далее индикатор). Пример подключения приведен для платы Arduino UNO. Коды программ приведены на языке С и проверены на указанной выше плате. К руководству прилагаются исходные коды простейших программ: «осциллограф» и «счетчик секунд».

Устройство индикатора и подключение к плате Arduino UNO.

Светящее поле индикатора состоит из 512 светодиодов (16x32). Изображение на индикаторе формируется динамически[1] — периодически, в определенный отрезок времени, информация отображается в отдельных частях индикатора. Для данного индикатора таких частей 4. Применение этого типа индикации позволяет существенно сократить количество соединительных проводов и уменьшить число ключевых элементов в схеме управления. При статической индикации, для управления N светодиодами, потребуется N источников тока - по одному на каждый светодиод и N проводников для соединения источников тока со светодиодами. В случае динамической индикации, светящее поле индикатора делится на M одинаковых частей, каждая часть коммутируется M ключами с N/M источниками тока. Динамический принцип индикации имеет недостатки — он требует постоянного переключения между отдельными частями изображения, что приводит к мерцанию светового поля при низких частотах переключений. Независимо от частоты переключений наблюдается другой эффект — за период перебора всех M частей изображения , каждый светодиод может светится только 1/M часть периода перебора, следовательно средняя за период излучаемая мощность уменьшается в M раз.

Рассмотрим схему индикатора (см. Приложение 1). Светодиоды D1 — D512 формируют светящее поле индикатора (16x32 светодиода). Светодиоды включены по матричной схеме [1] таким образом, что светятся при прохождении тока от линии ROW[n] (n = 0 — 3) — строки, к линии COL[m](m = 0 — 127) — столбцу. Из схемы видно, что физически индикатор организован как матрица 4x128 светодиодов (M = 4, N = 128).

Линии COL[m] организованы выходами 8-битных сдвиговых регистров DD2 — DD16 (номера битов BIT: 0 — 7, номера регистров REG[i](DD2 — DD16) i: 0 — 15). Микросхема 74HC595 содержит два регистра: сдвиговый и с параллельной загрузкой[2]. 16 регистров индикатора каскадированы и образуют один сдвиговый регистр длиной 128 бит[3].

Двоично-десятичный дешифратор (74HC138), управляет 4 силовыми ключами VT1 — VT4 и формирует линии ROW[n]. Значение n определяется комбинацией логических уровней на входах A и B(см. Таблица 2).

Назначение контактов разъема приведено в Таблице 1. Там же приведен один из вариантов соединения индикатора с платой Arduino UNO.

Для работы с индикатором используется аппаратный интерфейс SPI, представленный на плате Arduino UNO.

Номер контакта		Обозначение	Назначение сигнала
Индикатор	Arduino UNO		
1	2	-OE	выключение индикатора: 0; включение: 1
2	3	A	младший бит ROW
4	4	B	старший бит ROW
12	11	DATA	последовательные данные
10	5	LOAD	синхроимпульс записи в параллельный регистр
8	13	SCK	синхроимпульсы записи последовательных данных

Таблица 1. Таблица соединения контактов индикатора и платы Arduino UNO.

Логический уровень на контакте		Номер линии ROW
A	B	
0	0	0
1	0	1
0	1	2
1	1	3

Таблица 2. Выбор линии ROW..

В Таблице 2 представлена карта памяти индикатора. ROW[n] выделены отдельными цветами, для различных n. COL[m] — реализованы битами регистров REG[i]. Номера i указаны в ячейках таблицы.

Алгоритм работы с индикатором

Алгоритм работы с индикатором построен на периодическом вызове одной функции update(). При каждом вызове индикатор выводит часть изображения. Изображение должно быть размещено в части памяти контроллера — буфере индикатора.

Буфер индикатора может представлять собой двумерный массив 16 x 32 бит или 16 x 4 байта. Для удобной работы с битами, буфер индикатора можно представить в виде одномерного массива из 16 элементов 32-битных целых (unsigned long).

Для формирования изображения на индикаторе на определенные моменты времени, в регистры индикатора выводятся фрагменты каждой строки матрицы светодиодов (на карте памяти эти фрагменты помечены отдельными цветами) — последовательно вывести 16 байт в SPI-порт.

В Таблице 3 представлены последовательности для вывода двух строк индикатора, где n — номер элемента буфера индикатора, а m— номер бита, с которого выводится последовательность из 8-ми бит(байт) в SPI-порт.

col	regn	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	n	15	11	7	3	15	11	7	3	15	11	7	3	15	11	7	3
	m	24	24	24	24	16	16	16	16	8	8	8	8	0	0	0	0
1	n	14	10	6	2	14	10	6	2	14	10	6	2	14	10	6	2
	m	24	24	24	24	16	16	16	16	8	8	8	8	0	0	0	0

Таблица 3. Пример вывода двух строк индикатора

Значения n и m могут быть расчитаны по формулам:

n = 15-(i mod 4)*4-row (1.1)

m = 24-(i / 4)*8 (1.2)

В SPI-порт выводится data :

data = buf[n]>>m (2) ,

где buf — буфер индикатора.

Исходный код функции upload() представлен в Листинг 1.

```
/**
 * Вспомогательная функция:
 * Вычисление значения data
 * @param row – номер строки
 * @param buf – указатель на буфер индикатора
 * @param regn – номер регистра индикатора
*/
int get_img_byte(unsigned long* buf, int row , int
regn )
{
 int m = 24 - ( ( regn / 4 ) * 8 );
 int n = ( 15 - ( ( regn % 4 ) * 4 ) - row );
 return buf[n]>> m;
}
/**
 * Вспомогательная функция: Переключение строк
 * @param row – номер строки
*/
```

```
void set_row( int row )
{
  // Приводим row к нужному виду
  row &= 0x03;
  row = 3 - row;
  // Отключаем индикацию
  digitalWrite( pin_OE , LOW );
  // Переключаем ножк дешифратора
  digitalWrite( pin_A , ( row & 1 ) && HIGH );
  digitalWrite( pin_B , ( row & 2 ) && HIGH );
  // Выводим синхроимпульс регистра с
параллельной загрузкой
  digitalWrite( pin_LD , HIGH );
  digitalWrite( pin_LD , LOW );
  // Включаем индикацию
  digitalWrite( pin_OE , HIGH );
}
/**
 * Полный цикл обновления индикатора
 * @param buf — указатель на буфер индикатора
 */
void upload( unsigned long* buf )
{
 int regn, row;
 for ( row = 0 ; row < 4 ; row++ )
 {
  for( regn = 0 ; regn < 16 ; regn++ )
   SPI.transfer( ~get_img_byte( buf , row , regn ) );
  wr_clk( row );
 }
}
```

Листинг 1. Функции обновления индикатора.

Приложение 1. Упрощенная схема индикатора

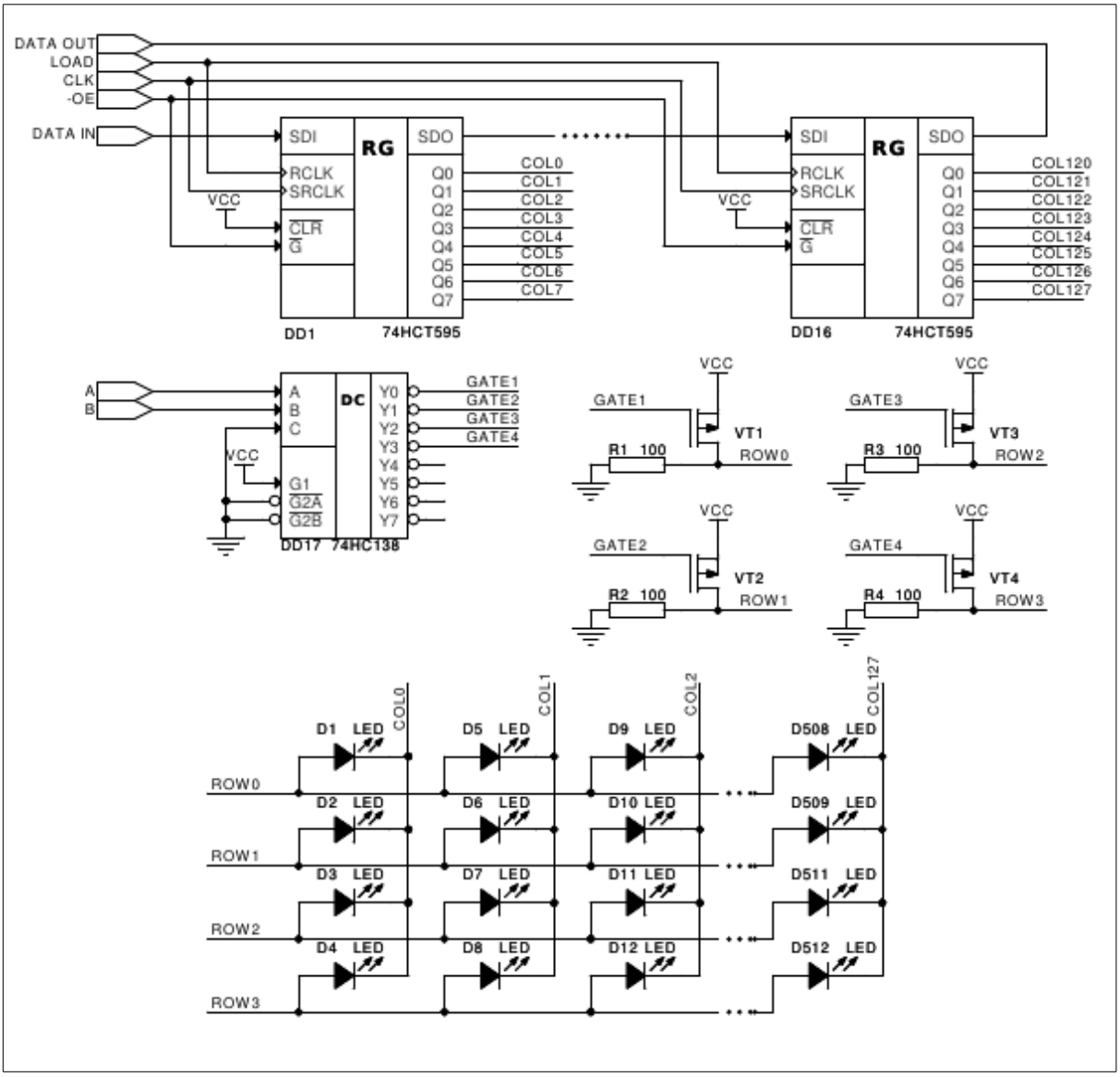

Рис. П1. Упрощенная схема индикатора.

Приложение 2. Карта памяти индикатора

X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
COL	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
BIT	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
m	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ROW	REG				Y	n
3	3	7	11	15	15	0
2	3	7	11	15	14	1
1	3	7	11	15	13	2
0	3	7	11	15	12	3
3	2	6	10	14	11	4
2	2	6	10	14	10	5
1	2	6	10	14	9	6
0	2	6	10	14	8	7
3	1	5	10	13	7	8
2	1	5	10	13	6	9
1	1	5	10	13	5	10
0	1	5	10	13	4	11
3	0	4	8	12	3	12
2	0	4	8	12	2	13
1	0	4	8	12	1	14
0	0	4	8	12	0	15

Таблица П1. Карта памяти индикатора.

Литература.

1. http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D0%B4%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80
2. http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf
3. http://arduino.ru/Tutorial/registr_74HC595

Гронский П.В.
Москва, 2013г.