

Символьные COG индикаторы

Пример реализации символьного индикатора на базе COG графики RDX0032

Евгений Рахно, ведущий специалист ООО «Гамма»

E-mail: info@microchip.ua

COG технология изготовления монохромных графических индикаторов позволяет полностью отказаться от «стандартных» графических индикаторов с разрешением вплоть до 132 × 64 и при этом стоимость COG индикатора примерно в два раза ниже стоимости «стандартной» графики. Именно с этим связана невероятная популярность данного типа индикаторов. И именно поэтому мы все чаще получаем вопросы о символьных COG индикаторах. Данная статья поможет вам найти ответы на ваши вопросы.

Прежде всего, давайте разберемся, что же такое символьный индикатор? Символьный индикатор формата 16×2 представляет собой две строки по 16 символов в каждой строке. Каждый символ представляет собой знакоместо 5×8 точек, при этом в формировании стандартного символа принимает участие поле с размерами 5×7 точек, а «нижний» ряд 5×1 служит для отображения «подчеркивания». Зазоры между точками минимальны, однако присутствуют интервалы между соседними символами. Эти интервалы являются «физическими» и заданы фирмой-производителем в процессе формирования стекла (рис. 1).

Теперь давайте поговорим о COG индикаторах.

COG индикатор состоит из двух стекол с электродами, между которыми находятся жидкие кристаллы. На поверхности одного из стекол при помощи специального раствора закреплен управляющий контроллер (отсюда и аббревиатура COG — Chip On Glass — чип на стекле). Для связи с микроконтроллером, а так же для крепления на печатную плату используются металлизированные выводы. Видимая область графического индикатора состоит из

«точек» прямоугольной либо квадратной формы. Зазоры между соседними точками минимальны.

Из чего состоит цена COG индикатора? Стоимость индикатора складывается из стоимости изготовления стекла с нанесением электродов, стоимости жидких кристаллов, стоимости управляющего контроллера и стоимости металлизированных выводов.

Что общего у графического и символьного COG индикаторов с одинаковыми габаритными размерами? Общее — площадь стекла, количество

жидких кристаллов, количество выводов, количество электродов. Для изготовления символьного индикатора 16×2 нам необходимо разбить площадь на 1280 точек (16×2 = 32 символа = 32×5×8 = 1280 точек). Для графики 128×32 это значение составляет 4096. Другими словами, сложность, а значит, и стоимость изготовления стекол одинаковы (сравните с сегментными COG индикаторами, где количество элементов составляет минимум 40 и максимум 120).

А что разное? Разное — контроллер. В случае с символьным индикатором управляющий контроллер должен иметь, переводя на понятный нам язык, 160 байт ОЗУ для хранения образа рабочего поля и 1280 байт ЕЕПРОМ для энергонезависимого знакогенератора. Контроллер графического индикатора должен иметь минимум 512 байт ОЗУ и может не содержать ЕЕПРОМ. Грубо говоря, стоимость контроллера для символьного индикатора примерно равна стоимости контроллера для графики.

И что у нас получается в итоге? Получается то, что стоимость символьного COG индикатора будет практически такой же, как стоимость графического COG индикатора. Так стоит ли вкладывать в разработку символьный COG индикатор, если графический COG стоит столько же?

Некоторые разработчики отвечают — да, стоит, так как для работы с графикой требуется более производительный контроллер с большим объемом памяти данных.

Вот именно этот момент хочется рассмотреть подробнее.

Как происходит общение с «обычным» символьным индикатором?

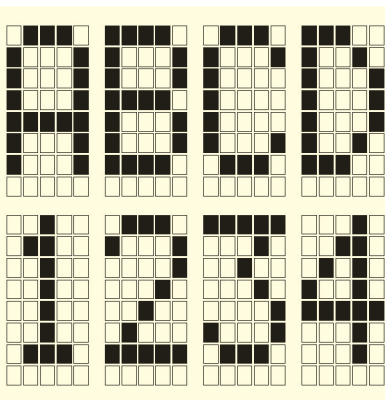


Рис. 1. «Стандартный» символьный индикатор

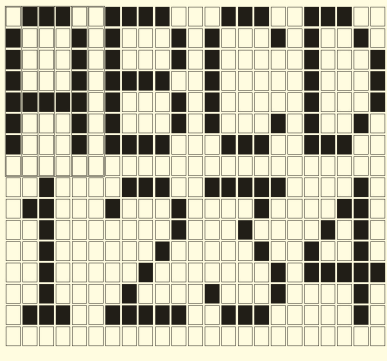


Рис. 2. формирование символов на экране графического индикатора

Практически все символьные индикаторы, присутствующие на рынке Украины, имеют драйвер ks0066u либо его аналог. Существует две схемы подключения данного драйвера к микроконтроллеру: «полная» — RS, RW, E, <D0..D7> (используется 11 портов ввода-вывода, данные передаются байтами) и «сокращенная» — RS, RW, E, <D4..D7> (используется 7 портов ввода-вывода, данные передаются тетрадами). Другими словами, если у микроконтроллера меньше семи портов ввода-вывода, то подключить индикатор к такому контроллеру вы не сможете (существует схема включения с 6-ю портами, однако автор данной статьи считает ее потенциально опасной).

Как происходит инициализация и работа с символьным индикатором?

Стандартно разработчик пишет несколько подпрограмм:

- *LCD_Init* — в эту подпрограмму помещаются команды включения ЖКИ, очистка его от «мусора», а также в ней реализуется выбор 4/8 битного интерфейса, задается режим инкремента адреса и т. д.;
- *LCD_set_addr* — выбор строки (0..1)/ столбца (0..15) индикатора для последующей записи;
- *LCD_puts_char* — вывод символа на экран ЖКИ;
- *LCD_puts_string* — вывод строки символов на экран ЖКИ (как правило, используется указатель на строку и функция *LCD_puts_char*).

Теперь давайте рассмотрим работу с графическим COG индикатором на примере популярного **RDX0032**.

RDX0032 имеет на борту I²C, а это значит, что для общения с ним нам необходимо всего два вывода микроконтроллера. Многие современные микроконтроллеры имеют I²C, а в случае программной реализации I²C ин-

дикатором RDX0032 сможет управлять даже «шестиногий» PIC10F200 (в случае с символьным COG индикатором вам также понадобилось бы всего две «ноги»).

Индикатор RDX0032 имеет разрешение 128×32 точки, при этом 8 точек по вертикали объединяется в один байт (рис. 2). Таким образом графический индикатор RDX0032 можно представить в виде двумерного массива 128×4.

Программное обеспечение:

- *COG_Init* — служит для начальной инициализации графического COG индикатора. Первое отличие COG графики от символьного индикатора заключается в отсутствии четкой привязки к «верху» и «низу», «лево» и «право». Формально, «низ» индикатора может быть определен из документации, однако на практике может оказаться так, что для удобства дизайна PCB «низ» должен быть вверху. Для этого случая в драйвере индикатора RDX0032 предусмотрен режим «отзеркаливания» по X и по Y. Второе отличие заключается в программной настройке контрастности — разработчик может предусмотреть пункт меню, в котором пользователь сможет подобрать контрастность в зависимости от уровня освещенности помещения, в котором эксплуатируется прибор (стандартный символьный индикатор этой возможности не имеет, а контрастность задается лишь аппаратно при помощи переменного резистора);
- *COG_set_addr* — выбор координаты X (0–127) и координаты Y (0–3) для последующей записи;
- *COG_puts_char* — вывод символа на экран графического COG индикатора;
- *COG_puts_string* — вывод строки символов на экран COG индикатора (как правило, используется указатель на строку, но для увеличения быстродействия вместо *COG_puts_char* для последовательного вывода символов желательно использовать последовательный вывод n-ного количества байт. Это связано с тем, что передача по I²C требует старта, стопа и адреса для каждой передачи, что приводит к увеличению количества передаваемых данных).

Для опытного программиста не составит особого труда реализовать описанные выше функции как для символьного, так и для графического индикатора.

Главное отличие RDX0032 от любого символьного индикатора заключается в отсутствии встроенного знакогенератора.

С одной стороны, это не очень удобно и отпугивает разработчиков, но, с другой стороны, пользователь может «изготовить» шрифт под себя.

Давайте рассмотрим несколько вариантов реализации знакогенератора на примере микроконтроллера семейства PIC16Fxxx.

В отличие от символьных индикаторов графика не имеет «интервалов» и это необходимо учесть. В простейшем случае одно знакоместо будет занимать площадь 6×8 (сам символ занимает поле 5×7, точки снизу и справа служат для обеспечения требуемого интервала).

Микроконтроллеры семейства PIC16 имеют 14-битное слово и некоторые из них позволяют производить чтение из Flash-памяти программ.

В этом случае целесообразно разбить один символ на три байта, а недостающий бит заменить на логический «0», тем самым сформировав интервал. Таким образом, таблица из 256 символов будет занимать 768 слов программы.

В том случае, если ваш микроконтроллер не поддерживает чтение из Flash, можно создать таблицу при помощи команды ассемблера RETLW (либо использовать соответствующие директивы C компилятора), например:

```
MOVWF FSR
MOVF INDF, W
RETURN
RETLW 0x7E
RETLW 0x88
RETLW 0x88
RETLW 0x88
RETLW 0x7E
RETLW 0xFE
RETLW 0x92
RETLW 0x92
RETLW 0x92
RETLW 0x6C
```

В этом случае таблица займет 256×5 = 1280 слов программы. На практике вам вряд ли понадобятся все 256 символов, а значит количество памяти, отведенной под таблицу, будет меньше.

Так же вам понадобится небольшая таблица перекодировки для работы с символами в явном виде.

Приведем пример вывода символа «А» в левый нижний угол экрана:

```

void COG_puts_char (uint8 c){
i2c_start();           // формирование сигнала "старт"
i2c_write(0b01110010); // адрес индикатора RDX032
i2c_write(font[c][0]); //вывод первого столбца символа
i2c_write(font[c][1]); //вывод второго столбца символа
i2c_write(font[c][2]); //вывод третьего столбца символа
i2c_write(font[c][3]); //вывод четвертого столбца символа
i2c_write(font[c][4]); //вывод пятого столбца символа
i2c_write(0x00);       // формирование "интервала" между символами
i2c_stop();            // формирование сигнала "стоп"
}

```

Рис. 3. Функция COG_puts_char

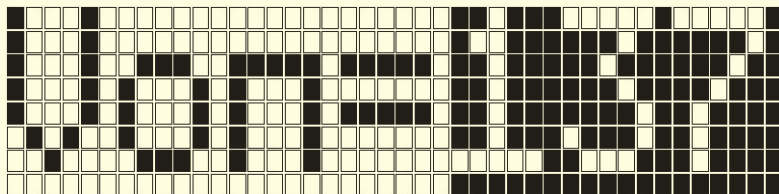


Рис. 4. Ввод значения опорного напряжения

```

COG_set_addr (0, 0); // выбор
                  // координаты
COG_puts_char ('A'); // вывод
                  // символа

```

где функция `COG_puts_char` имеет вид, приведенный на рисунке 3.

Ну вот, собственно, и все — мы получили аналог символьного индикатора на основе COG графики RDX0032. Теперь достаточно заменить функции `LCD_Init`, `LCD_puts_char` и т.д. на `COG_Init`, `COG_puts_char` и дальнейшая модернизация кода программы не понадобится (пример реализации знакогенераторов можно получить, отправив запрос на info@microchip.ua).

А теперь расширим дополнительные возможности «символьного» индикатора RDX0032 без использования каких-либо графических примитивов:

- **Инверсия символов и слов** (рис. 4). Данный прием может применяться для акцентирования внимания на каком-либо объекте. Типичный пример применения — выделение редактируемого параметра. Метод реализации — инверсия значений таблицы символов во время вывода на экран.
- **Мигание**. Используется для привлечения внимания. Типичный пример применения — параметр вышел за пределы рабочего диапазона.
- **Горизонтальная бегущая строка** либо **горизонтальное смещение** экрана. В отличие от символьного индикатора, где пользователю доступно лишь 16 значений координаты

X, графика позволяет делать вывод первого символа в точке с координатами (0, 0), (1, 0), (2, 0), (3, 0), (4, 0) и, наконец, (5, 0). Таким образом мы можем получить плавное смещение текста (при реализации бегущей строки на обычном индикаторе мы можем довольствоваться лишь скачкообразным перемещением символов).

- **Вертикальное смещение экрана**. Алгоритм достаточно прост — после начальной инициализации экрана считываем 8 байт первой строки (или сколько не жалко памяти), далее — 8 байт второй, третьей и четвертой, а затем осуществляем сдвиг на один бит и записываем полученные данные обратно.
- **Вертикальное смещение символа**. Используя данный нехитрый прием мы можем получить эффект механического счетчика (рис. 5).
- **Шрифт x2** (рис. 6). Индикатор RDX0032 имеет разрешение 128×32 и используя знакогенератор с базовым размером 6×8 мы можем вывести на экран 4 строки по 21 символу в строке. Используя масштабирование таблицы символов, легко получить символ с размерами 12×16 (размер символа 10×14, интервал по X две точки, 2 строки по 10 символов), либо 11×16 (размер символа 10×14, интервал по X одна точка, 2 строки по 11 символов).
- 7. **Шрифт x4**. В исключительных случаях для вывода числовых показаний можно использовать масштаб

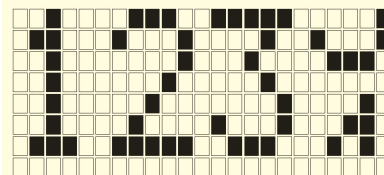


Рис. 5. «Механический счетчик» — написать не сложно, а выглядит эффектно, особенно в динамике

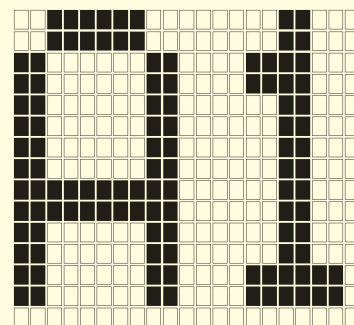


Рис. 6. Масштабирование базового шрифта

x4, однако имея в запасе высоту 32 символа, Вы вряд ли захотите это делать... Необходимо отметить, что масштаб текста, как и сам знакогенератор, являются условностью. Вы запросто можете использовать «одинарный» шрифт в верхнем левом углу индикатора, и, одновременно, использовать масштаб x2 в центральной его части, таким образом отделив важную информацию от второстепенной. Либо же вы можете составить свой собственный знакогенератор, включив в него интересные вас символы, к примеру, °C, или составить отдельный знакогенератор 10×14 для цифр.

Вот далеко не полный перечень дополнительных возможностей, предоставляемых разработчику графическим индикатором RDX0032, и при этом мы даже не затрагивали основную функцию данного индикатора — отрисовку графики... Так стоит ли ограничивать себя, закладывая в разработку символьный COG индикатор?

Более детальную информацию можно получить у специалистов ООО «Гамма»:

**тел.: (056) 745-46-54,
(056) 745-46-65, (066) 173-26-79,
(096) 480-38-65, (0562) 36-09-41,
(0562) 36-07-92, (044) 494-35-72,
e-mail: info@microchip.ua,
<http://www.microchip.ua> **