



Рис. 4. 3D-модель в DipTrace печатной платы четырёхканального ретранслятора

Передачу сигнала выполняет управляемый транзистор Q1(2N7000), который замыкает шину 1-Wire к уровню 0 В и тем самым формирует из него информационный сигнал нуля или единицы стандарта 1-Wire для приёма на компьютере.

Временные интервалы замыкания формирует МК U1(12F629). При использовании внутреннего RC генератора в режиме работы 4 МГц погрешность сигналов управления составляет  $\pm 1$  мкс, что не вызывает ошибок в сигналах при приёме и идентификации данных интерфейса 1-Wire.

Необходимо учесть дополнительный потребляемый ток МК в 1...3 мА по шине +5 В и возможное падение напряжения для длинной шины питания.

## Конструкция

Схема и все компоненты собраны на печатной плате и предназначены для монтажа «под винт». На рис. 4 представлена 3D-модель платы. Печатная плата разработана в среде проектирования DipTrace, проект доступен в каталоге программ [3] и находится в файле **sh\_12F629\_reg1\_v1.zip**.

Предусмотрена установка транзистора в разных исполнениях TO-92 и SOT-23. МК устанавливается через переходную колодку DIN-8, что позволяет модифицировать и изменять программное обеспечение и возможные функции использования.

## Программное обеспечение

Для работы схемы ретранслятора используется программное обеспечение, совместимое с интерфейсом 1-Wire, которое представлено в [3]. Это программа термометрии **silos\_v3.exe**, подготовленная на языке программирования FORTH [4]. Программа содержит все компоненты для программирования и модификации режимов работы

МК 12F629 [2], а также печатные платы и схемы.

Программный код формируется в режиме «восстановление всех файлов – компиляция нового кода» [5] и доступен в [3], файл **silos\_v3.exe**.

Рассмотрим несколько поясняющих примеров функционирования программного обеспечения для схемы ретранслятора на 12F629 на языке программирования FORTH. Текст программы находится в файле **silos\_v3\_12f629\_regn\_v1.f** и компилируется через выполнение общей программы **silos\_v3.f** с помощью файла **100\_spf4.exe**. Для уточнения, где описание и где программа, фрагменты текста программ, «вырезанные» из основного текста, размещены в цветных врезках.

Внимание: так как компиляция занимает время, подключение библиотек для приведённых других авторских устройств в данной программе переведено в комментарий – косая черта влево. При компиляции сразу нескольких библиотек их состав «пересекается», и в этом случае будет выдаваться сообщение о дублировании форт-слов. Дублирование форт-слов создаёт избыточность кода и является источником ошибок в работе основного алгоритма программы сбора данных термометрии.

### Пример 1. Запрос по шине ретранслятора к датчикам с использованием косвенной адресации

Данные о номере датчика передаются от главного ПК и записываются в ячейки с названием NUM\_BT0...7.

```
0x2B equ NUM_BT0 \ нач. байт приёма данных - 9300080100D00528, итого 8 регистров
```

```
* * * \ регистры от 0x2B до 0x32
0x32 equ NUM_BT7 \ конечный байт
```

Для обращения к данным для чтения и записи восьми байт номера датчика и побитной передачи-приёма в канал ретрансляции используются регистры косвенной адресации.

```
0x04 equ FSR \ адрес косвенного регистра для работ с INDF
```

```
0x00 equ INDF \ регистр косвенной адресации
```

Текущий используемый номер регистра находится в FSR, а действия с ним выполняются в INDF.

Рассмотрим подпрограмму **WORKS\_DAN\_MAIN\_SBORS\_NUM\_BT#** передачи сигнала сброса RST в канал MAIN ретранслятора с фрагментом ретран-

сляции номера термодатчика DS1820 методом косвенной адресации.

```
\ выдан сигнал RST и код байта 0x55, назначить номер датчика.
Далее выдаём этот номер
0x2B movlw \ в акм. первый байт данных хранения номера
FSR movwf \ передали в косвенный регистр значение регистра первого байта
0x8 movlw \ количество байт в номере, т.е. 8 байт
N_1W8 movwf \ передали в счётчик учёта байт
ORG @ m1 ! INDF movf \ из регистра в акм. Передали байт
1 MAIN_OUT_8BT# \ подпрограмма передачи байта в шину main из акм.
0x80 movlw \ значение паузы на 50 мкс между байтами
1 PAUSE_1W# \ выполняем паузу через подпрограмму
0x1 movlw \ единица для увеличения номера регистра байта данных
FSR 0xF addwf \ складываем адрес FSR+1, т.е. переходим к сл. байту.
N_1W8 0xF decfsz \ вычитаем 1 из счётчика байт и пропускаем, когда 0. Т.е. выйти из пп.
m1 goto \ повторяем следующие байты, если они не закончились.
```

Таким образом, косвенная адресация позволяет циклически обращаться к регистрам хранения данных. В данном примере полученные данные ретранслируются далее к датчику.

Необходимо обратить внимание на паузу между передачами байта. С шиной +5 В можно от паузы отказаться. В этом случае пакет идёт «плотно». Если используется двухпроводный режим с паразитным питанием, то без паузы этого питания не хватает для работы датчика DS1820, поскольку сопротивление подтяжки в МК 12F629 составляет около 10 кОм.

### Пример 2. Передача кода формата 1-Wire в шину ретранслятора

Передаётся байт из аккумулятора в шину ретранслятора в подпрограмме форт-слова **MAIN\_OUT\_8BT#**.

```
BYAT_1W movwf \ данные в регистр из акм. Передача бита=0
```

```
0x8 movlw \ количество бит в байте, т.е. кол-во циклов
```

```
N_1W movwf \ счётчик загрузили
ORG @ m1 ! porta 0x5 bcf \ GP5 установили в 0 - это вывод ретранслятора MAIN
```

```
BYAT_1W 0x0 btfsf \ проверка на 0 b=0. Пропустить, если =0
```