Программируемые системы на кристалле

Часть 2. Графический генератор приложений PSoC Express

Продолжаем рассказ о программируемых системах на кристалле PSoC Cypress. В данной статье вы познакомитесь с новым графическим редактором приложений PSoC Express. Даже если вы никогда в жизни не программировали микроконтроллеры и никогда не сталкивались с PSoC, вам может быть интересен и полезен данный материал, поскольку по его прочтении вы сможете делать несложные электронные устройства на базе микроконтроллера, не читая при этом его описание и не зная ни одного языка программирования.

Александр Кузминский

Alexander.Kuzminsky@macrogroup.ru

Для начала вкратце напомним о том, что такое PSoC и чем он полезен. Микросхема PSoC (Programmable System on Chip) компании Cypress является микроконтроллером с встроенным массивом аналого-цифровых ресурсов. Благодаря этому внутри PSoC можно реализовать обработку как аналоговых, так и цифровых сигналов. Обычно эти микросхемы используются в промышленной автоматике, охранных системах, бытовой и автоэлектронике.

В рамках данной статьи мы специально не будем углубляться в архитектуру PSoC (см. «КиТ» № 4'2005), а также в тонкости программирования. Более того, мы попробуем сделать несколько электронных устройств, практически не вникая в характеристики элементной базы. Это стало возможным благодаря появлению нового программного обеспечения PSoC Express (на момент написания статьи — версии 1.1).

PSoC Express

PSoC Express — бесплатно распространяемый генератор приложений, или, говоря проще, программа, которая позволяет создавать электронные устройства, используя только графический интерфейс. Пакет PSoC Express доступен для скачивания на сайте производителя (**ht tp:/ /ww w.cypress.com/psocexpress**), и может работать без каких-либо аппаратных средств.

Весь процесс проектирования условно можно разделить на 3 этапа:

1. **Выбор из библиотеки элементов, которые обеспечивают входные данные:**
   a. Датчики температуры (микросхемы Maxim, Fairchild, National);
   b. Терморезисторы с задаваемыми характеристиками;
   c. Цифровой вход (Pull Up, Pull Down, High Z);
   d. Входные напряжения в диапазонах 0–2,6 В, 0–5 В, 0–12 В, 0–31 В;
   e. Потенциометры;
   f. Кнопки, переключатели;
   g. Тахометры (импульсы с датчика скорости).
2. **Выбор из библиотеки элементов, которые обеспечивают выходные данные:**
   a. Аналоговые выходы 0–$U_{пит}$.
   b. Зуммеры;
   c. Ключи 10 мА (5 В), 5 А (5 В), 5 А (12 В), 10 А (48 В);
   d. ШИМ 10 мА (5 В), 5 А (5 В), 5 А (12 В), 10 А (48 В);
   e. Реле 5 В, 12 В, 24 В, 48 В;
   f. Светодиоды (в том числе мигающие и с регулируемой интенсивностью);
   g. Электродвигатели 3,3–48 В, 1–10 А (в том числе с регулируемой скоростью вращения).
3. **Задание функции зависимости выходных сигналов от входных.** Делается это либо посредством логических выражений (If… Then… Else…), либо с помощью таблицы соответствия.

После выполнения операций в соответствии с этими этапами пользователь может произвести программную симуляцию проекта. Убедившись в правильности сделанных построений, можно перейти в окно «BOM/Schematic» для получения полной информации о своем проекте. В результате этого действия программа PSoC Designer сгенерирует:

1. BOM (Bill Of Materials) — полный список компонентов, использованных в проекте.
2. Полное техническое описание на проект.
3. Принципиальная схема на проект.
4. Файл прошивки для микросхемы PSoC.

Теперь остается купить в магазине электронные компоненты согласно полученному списку, запаять их на плате по полученной принципиальной схеме и записать файл прошивки в микросхему PSoC (об этом подробно рассказывается в конце статьи). Все. Электронное устройство готово и при включении питания исправно заработает. При этом, заметьте, не потребовалось знания языков программирования Си или ассемблер, как, впрочем, и полного описания на PSoC.

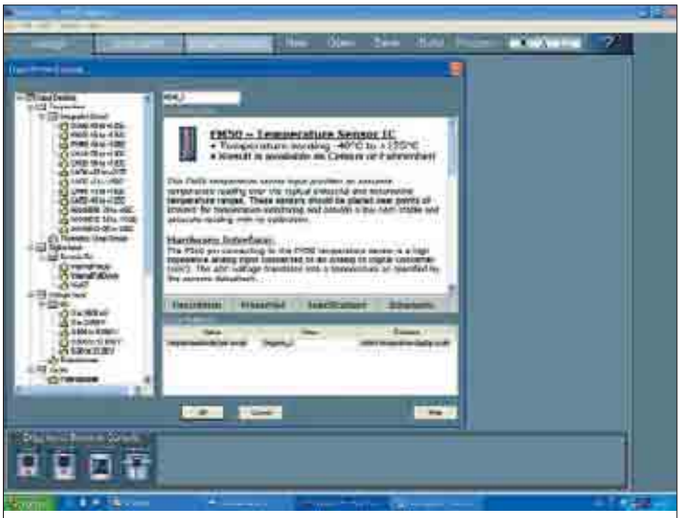

Рис. 1

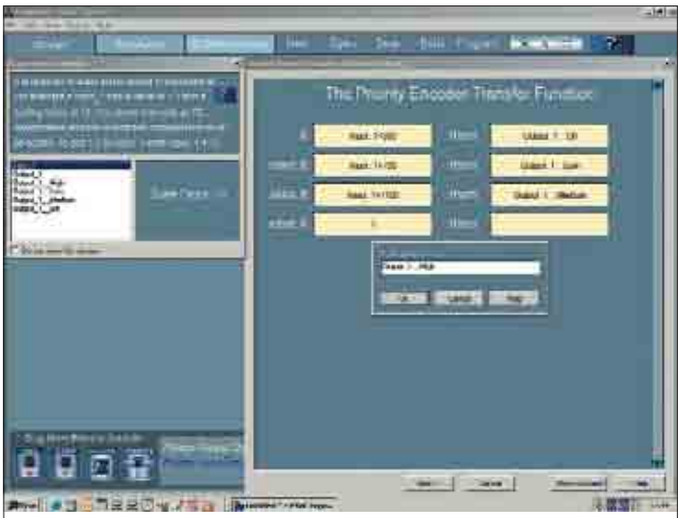

Рис. 2

Итак, чтобы внести ясность, покажем процесс создания некоторого электронного устройства. Например терморегулятора.

Терморегулятор

Предположим, мы хотим сделать устройство, управляющее скоростью вращения вентилятора в зависимости от температуры по следующему закону:

Таблица 1

Температура < 30 °C	Вентилятор выключен
30 °C ≤ температура < 70 °C	Вентилятор включен, скорость вращения минимальная
70 °C ≤ температура < 110 °C	Вентилятор включен, скорость вращения средняя
110 °C ≤ температура	Вентилятор включен, скорость вращения максимальная

Таким образом, в нашем проекте будет присутствовать датчик температуры, электродвигатель вентилятора, микросхема PSoC, управляющая работой вентилятора, и некоторые дискретные элементы.

Шаг 1. Запускаем PSoC Express (рис. 1). Появляется рабочее окно «Design» и окно помощи. Закроем последнее. В левой нижней части рабочего окна «Design» находятся 4 иконки:

- **Input** — входной сигнал.
- **Output** — выходной сигнал.
- **Valuator** — функция зависимости выходов от входов Output=f(Inputs).
- **Interface** — коммуникационный интерфейс I^2C.

Нажмем на иконку Input. В результате раскроется каталог устройств (рис. 1), разрешенных для установки на входе нашего изделия. В верхней части — список термодатчиков разных производителей. Выберем, к примеру, LM20 фирмы National Semiconductors. В правой части каталога можно просмотреть техническое описание и схему подключения для соответствующего элемента. Температурный датчик LM20 позволяет проводить измерения в диапазоне температур –55...+130 °C. Внутри PSoC реализован программный алгоритм, преобразующий данные с термодатчика в 16-битное целое число с дискретностью температуры 0,1 °C. Это означает, что значение температуры 117,9 °C будет сохранено во внутреннем регистре PSoC как 1179. Итак, нажав «OK», добавляем датчик температуры в наш проект. Видим, что на рабочем столе «Design» появилась соответствующая иконка с именем Input_1.

Шаг 2. Нажимаем на иконку Output. В раскрывшемся окне мы видим каталог устройств, разрешенных для установки на выходе нашего изделия. Спустившись в нижнюю часть каталога, можно увидеть двигатели разных типов. Выберем электродвигатель с номинальным рабочим напряжением 12 В и номинальным током 4 А. Управляющим элементом для двигателя служит MOSFET. (FAN→Brushless DC→Speed Control→External Drive Fet→12V 2-/3-Wire). В результате на рабочем столе появится вторая иконка, обозначенная как Output_1.

Шаг 3. Теперь необходимо задать связь между вентилятором и датчиком температуры, то есть определить функцию Output_1 = = Output_1(Input_1). Нажимаем правой кнопкой мыши на иконку Output_1 и в раскрывшемся окне выбираем Transfer Function. Функция преобразования (Transfer Function) бывает трех видов:

- Priority Encoder;
- Status Encoder;
- Table Lookup.

Не вдаваясь в ненужные пока подробности, выбираем первый тип (Priority Encoder). Эта функция позволяет задавать связь в форме логических условий: **if** (условие) **then** (действие).

Условие и действие нам необходимо задать. Рядом с окном Priority Encoder (рис. 2) открылось небольшое окно Expression Assistant — проще говоря, подсказка, с помощью которой можно увидеть имена переменных, которые могут участвовать в логическом выражении. Для каждой переменной присутствует необходимый комментарий. В частности, нажав в окне Expression Assistant строку Input_1, в комментарии мы увидим надпись «Scale Factor: 10». Это коэффициент масштабирования, о котором шла речь при описании свойств датчика температуры. Таким образом, если мы хотим указать температуру 30 °C, в логическом выражении мы должны написать 300, для температуры 70 °C нужно написать 700, а для температуры 110 °C — 1100. Итак, заполняем таблицу Priority Encoder следующим образом.

	if	Input_1<300	then	Output_1__Off
else	if	Input_1<700	then	Output_1__Low
else	if	Input_1<1100	then	Output_1__Medium
else	if	1	then	Output_1__High

Заметьте, что в последней строке в качестве условия стоит «1» — логическая единица. Это означает, что если ни одно из предыдущих условий выполнено не было, будет принято последнее. Таким образом можно уберечь себя от ошибки в случае неопределенных состояний.

Все! На этом процесс проектирования закончен. Все остальное — это задача PSoC Express. Последнее, что мы можем сделать, — это перейти на вкладку «Simulation» и проверить работу вентилятора во всем диапазоне температур (рис. 3).

Шаг 4. Нажмем кнопку «Built». В этот момент PSoC Express начинает создавать пакет файлов для вашего проекта, в том числе файл прошивки (*.hex) для микросхемы PSoC. Во время этого процесса вам будет предложено выбрать наиболее подходящий тип корпуса микросхемы PSoC. По окончании процесса PSoC Express создает документы:

- BOM;
- DataSheet;
- Schematic.

BOM

BOM (Bill Of Materials) — полный список компонентов, использованных в проекте.

На все компоненты приводится также номер в каталоге Digikey, с помощью которого на сайте w ww .digikey.c om можно посмотреть краткое описание и розничную цену на данный компонент.

DataSheet

В этом документе PSoC Express генерирует технические характеристики элементов, входящих в проект. В нашем примере здесь будет содержаться подробное описание датчика температуры и вентилятора.

Schematic

Фактически PSoC Express создает принципиальную схему (рис. 4), согласно которой вы должны осуществить стыковку всех компонентов устройства.

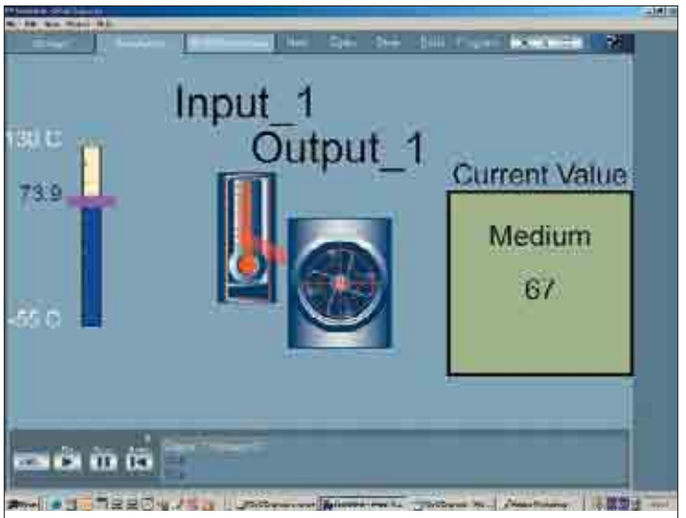

Рис. 3

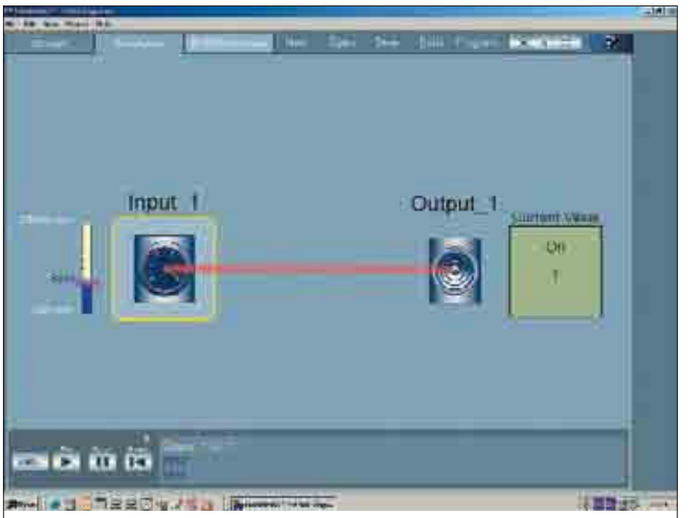

Рис. 5

Таблица 2

Label	Device	Package	Value	Digikey Part Number	Notes
Untitled_R1	Resistor	Radial	1k	1.0KQBK-ND	–
Untitled_J1	100mil	Straight	–	WM4203-ND	ISSP Connector
Untitled_S1	Button	–	–	P8011S-ND	Reset Switch – Normally Open
Untitled_C1	Capacitor	Radial	0,1uF	399-2155-ND	–
Untitled_U1	CY8C27443	PDIP-28	–	428-1596-ND	–
Input_1_U1	LM20BIM7	SC-70-5	–	LM20BIM7CT-ND	–55 °C to +130 °C
Input_1_C1	Capacitor	Radial	0,1uF	399-2155-ND	–
Input_1_C2	Capacitor	Radial	0,1uF	399-2155-ND	–
Output_1_R1	Resistor	Radial	1k	1.0KQBK-ND	–
Output_1_Q1	IRF7463	SO-8	–	IRF7463-ND	–
Output_1_M1	Fan	–	–	P11039-ND	–

Итак, что мы имеем? Список компонентов, схему и загрузочный код для PSoC Cypress. Мы создали законченное электронное устройство — терморегулятор, пользуясь только графическим интерфейсом.

Электронный ограничитель скорости

Это еще один пример создания готового электронного изделия с помощью графического генератора приложений PSoC Express.

Предположим, что устройство, которое мы хотим разработать, подключено к мотору или вращающемуся валу, с которого выходит сигнал с частотой, пропорциональной скорости вращения. Наша задача состоит в том, чтобы при превышении заданной предельной скорости вращения подать определенный звуковой сигнал. Попробуем сгенерировать такой звуковой сигнал с помощью зуммера. Таким образом, когда скорость вращения превышает предельный порог, предположим, 6000 об/мин, зуммер начинает пищать.

Шаг 1. Запускаем PSoC Express и нажимаем иконку Input. В раскрывшемся каталоге выбираем самое нижнее значение (Speed→Tachometer). В техническом описании на этот элемент видим, что он способен работать с частотами 300–25 000 Гц (или об/мин).

Шаг 2. Нажимаем иконку Output и выбираем в каталоге зуммер на 5 В и 100 мА (Audio→Buzzer→5V, 100mA drive). В результате на рабочем столе появится вторая иконка, обозначенная «Output_1».

Шаг 3. Нажимаем правой кнопкой мыши на иконку Output_1, и в раскрывшемся окне выбираем Transfer Function. Выбираем тип функции Status Encoder, и попадаем в уже знакомое из предыдущего примера окно. Для того чтобы правильно задать функцию связи выхода с входом, заполняем таблицу Status Encoder следующим образом.

```
if    Input_1<6000     then    Output_1__Off
if    Input_1>=6000    then    Output_1__On
```

Перейдя на вкладку «Simulation» (рис. 5), мы опять можем проверить правильность своих действий, меняя частоту оборотов и отслеживая состояние зуммера.

Шаг 4. Нажимаем кнопку «Built». В результате получаем полный список комплектующих, техническое описание, схему и файл прошивки.

Итак, мы сделали еще одно электронное устройство, затратив на это минимум времени и имея минимум знаний.

Остается лишь рассмотреть один немаловажный вопрос — как загрузить прошивочный файл (*.hex) в микросхему PSoC Cypress? Есть 2 варианта:

Вариант 1. Приобрести один из отладочных наборов, предлагаемых компанией Cypress. Подробный обзор таких наборов приведен в первой части публикации. Замечу лишь, что самое удобное — начать с набора CY3210miniprog стоимостью 860 руб. В составе набора кроме программатора есть 2 микросхемы PSoC Cypress, и последние не придется приобретать отдель-

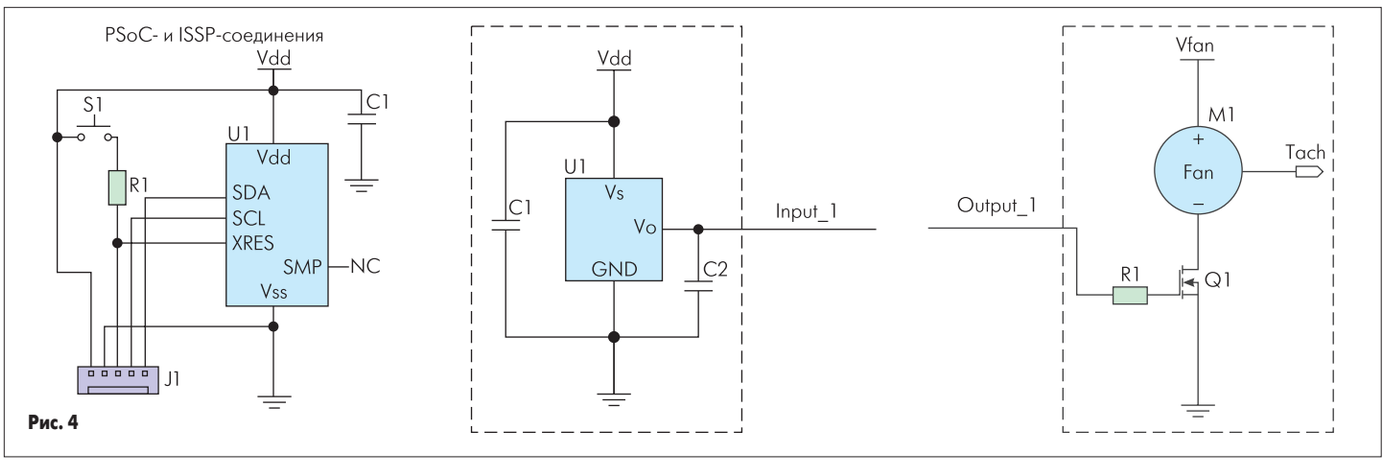

Рис. 4

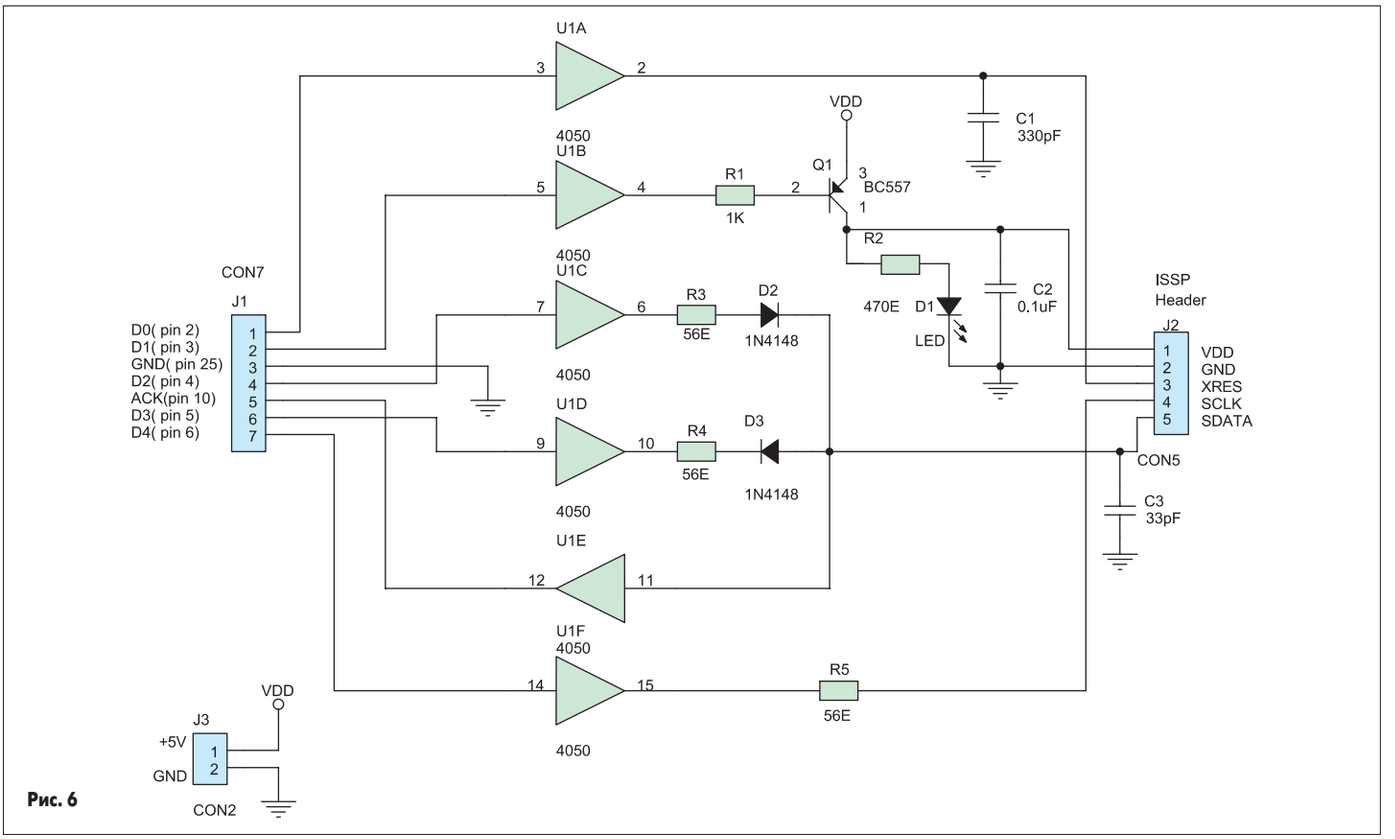

Рис. 6

но. Но если вы твердо решили обойтись минимумом средств, можете попробовать другой вариант.

Вариант 2. Программатор для PSoC можно сделать самому. К счастью, Cypress держит открытым протокол программирования и схему программатора (рис. 6). Все необходимое всегда можно найти в Интернете по адресу **ht tp:/ /w ww .macro-peterburg.ru/cypress/PSoC/PSoC_programmator.ht ml**.

В заключение хочется отметить, что графический генератор приложений PSoC Express — это наиболее простой способ начать работу с программируемыми системами на кристалле, даже если до этого момента вы никогда не работали с микроконтроллерами вообще. PSoC Express очень хорошо подходит для постановки лабораторных работ в вузах и помощи студентам в освоении современной элементной базы. Для опытных пользователей существует более мощный и одновременно более сложный пакет PSoC Designer, позволяющий перейти на более детальный уровень проектирования, но требующий значительных знаний по программированию.