



# ОГЛАВЛЕНИЕ

Принятые сокращения .....	12
Предисловие .....	13
Введение .....	14
<b>Глава 1. Параметры системы</b> .....	<b>15</b>
1.1. Динамический диапазон .....	15
1.2. Точность измерения напряжения .....	16
1.3. Калибровка .....	16
1.4. Пропускная способность шины .....	19
1.5. Производительность процессора .....	20
1.5.1. Прерывания .....	21
1.5.2. Интерфейсы .....	21
1.5.3. Поддержка на аппаратном уровне .....	21
1.5.4. Требования к процессору .....	22
1.5.5. Требования к операционной системе .....	22
1.5.6. Язык программирования и компилятор .....	22
1.6. Ограничение скорости .....	22
1.6.1. Затраты .....	23
1.6.2. Электромагнитная совместимость .....	23
1.7. Другие системные ограничения .....	24
1.7.1. Периферийные устройства .....	24
1.7.2. Общие интерфейсы .....	24
1.7.3. Приоритет задач .....	24
1.7.4. Системные требования .....	25
1.7.5. Разрядность цифрового слова .....	27
1.7.6. Интерфейсы .....	27
1.8. Частота дискретизации и наложение .....	27
<b>Глава 2. Аналого-цифровые преобразователи</b> .....	<b>30</b>
2.1. Общие сведения .....	30
2.2. Описание АЦП .....	32
2.2.1. Опорное напряжение .....	33
2.2.2. Выходное слово .....	33

2.2.3. Разрешение	33
2.3. Типы АЦП	34
2.3.1. Следящий АЦП	34
2.3.2. Параллельный АЦП	36
2.3.3. АЦП последовательных приближений	37
2.3.4. АЦП двойного интегрирования	38
2.3.5. Сигма-дельта АЦП	40
2.3.6. Составной АЦП	43
2.4. Сравнение типов АЦП по основным показателям	44
2.5. Устройства выборки-хранения	45
2.6. Реальные компоненты	48
2.6.1. Входные уровни	48
2.6.2. Встроенный источник опорного напряжения	49
2.6.3. Дополнительная емкость на входе опорного напряжения	49
2.6.4. Встроенное УВХ	49
2.7. Интерфейс микропроцессора	49
2.7.1. Кодирование выходного слова	49
2.7.2. Параллельный интерфейс	50
2.7.3. Время доступа	52
2.7.4. Выход BUSY	54
2.7.5. Время освобождения шины	54
2.7.6. Связь сигналов шины управления с АЦП	55
2.7.7. Задержка между преобразованиями	55
2.7.8. Величина погрешности преобразования	56
2.8. Синхронные интерфейсы	56
2.9. Последовательные интерфейсы	58
2.9.1. Периферийный последовательный интерфейс SPI/Microwire	58
2.9.2. Шина I <sup>2</sup> C	61
2.9.3. Шина SMBus	63
2.9.4. Специализированные последовательные интерфейсы	64
2.10. Многоканальные АЦП	65
2.11. Встроенные в микроконтроллер АЦП	65
2.12. Кодеки	67
2.13. Прерывания	67
2.14. Многофункциональные выходы микроконтроллеров	68
2.15. В рабочую тетрадь разработчика	70
<b>Глава 3. Датчики</b>	<b>71</b>
3.1. Температурные датчики	71
3.1.1. Терморезисторы	71
3.1.2. Масштабирование	75
3.1.3. Влияние разброса параметров компонентов	79

3.1.4. Резистивные температурные датчики . . . . .	81
3.1.5. Термопары . . . . .	81
3.1.6. Полупроводниковые температурные датчики . . . . .	84
3.2. Оптические датчики . . . . .	85
3.2.1. Щелевой оптрон . . . . .	85
3.2.2. Отражательный оптрон . . . . .	85
3.2.3. Особенности работы в ИК-диапазоне. . . . .	88
3.2.4. Механическая нестабильность. . . . .	90
3.2.5. Датчик защитного кожуха . . . . .	91
3.2.6. Управление несколькими датчиками . . . . .	92
3.2.7. Оптрон с закрытым оптическим каналом . . . . .	94
3.2.8. Дискретные оптические датчики . . . . .	96
3.3. Приборы с зарядовой связью . . . . .	99
3.3.1. Основы работы ПЗС . . . . .	99
3.3.2. Управление экспозицией . . . . .	101
3.3.3. ПЗС-линейки . . . . .	101
3.3.4. Распознавание цвета . . . . .	103
3.3.5. Трехлинейные ПЗС . . . . .	104
3.3.6. Методы цветовой обработки . . . . .	106
3.3.7. Чувствительная площадь ПЗС-матрицы . . . . .	106
3.3.8. Темновой заряд . . . . .	107
3.3.9. Двойная коррелированная выборка . . . . .	107
3.3.10. Неравномерность . . . . .	108
3.3.11. Питание ПЗС . . . . .	110
3.3.12. АЦП ПЗС . . . . .	110
3.4. Датчики магнитного поля . . . . .	110
3.4.1. Датчики Холла . . . . .	110
3.4.2. Линейные дифференциальные трансформаторы . . . . .	113
3.4.3. Датчики с переменным магнитным сопротивлением . . . . .	114
3.5. Полупроводниковые тензодатчики движения/ускорения . . . . .	116
3.6. Металлические тензодатчики веса/силы. . . . .	117
3.7. Ключи . . . . .	119
<b>Глава 4. Измерение временных параметров . . . . .</b>	<b>121</b>
4.1. Общие сведения . . . . .	121
4.2. Измерение периода или частоты. . . . .	125
4.3. Смещение частот . . . . .	127
4.4. Преобразователи напряжения-частота . . . . .	129
4.4.1. Применение ПНЧ . . . . .	130
4.4.2. Фильтрация . . . . .	131
4.5. Тактовая частота и диапазон . . . . .	131
4.6. Увеличение точности при ограниченном разрешении . . . . .	133

<b>Глава 5. Методы управления</b> .....	139
5.1. Управление с разомкнутой петлей обратной связи .....	139
5.2. Отрицательная обратная связь .....	139
5.3. Микропроцессорные системы управления .....	140
5.4. Система релейного управления .....	141
5.5. Температурный перегрев .....	144
5.6. Система пропорционального управления .....	144
5.7. Пропорционально-интегрально-дифференциальное управление .....	148
5.7.1. Вклад операции дифференцирования в алгоритм управления .....	151
5.7.2. Вклад операции интегрирования в алгоритм управления .....	153
5.7.3. ПИД-управление в целом .....	155
5.7.4. Способы применения системы ПИД-управления .....	156
5.7.5. Насыщение .....	157
5.7.6. Программное обеспечение .....	159
5.7.7. Временная задержка .....	159
5.7.8. Резкие изменения сигнала .....	160
5.7.9. Специальные требования .....	161
5.8. Управление двигателем .....	161
5.8.1. Пример постоянной скорости вращения круговой карусели .....	163
5.8.2. Позиционирование .....	165
5.8.3. Программное обеспечение .....	167
5.9. Управление с предсказанием .....	168
5.10. Измерение и анализ реакции петли обратной связи .....	169
5.10.1. Логический анализатор/Цифровой осциллограф .....	170
5.10.2. Измерение параметров двигателя .....	171
5.10.3. Доступное программное обеспечение .....	173
5.11. Примеры программирования ПИД-систем .....	173
5.11.1. Основная петля ПИД-управления .....	173
5.11.2. Функция запрета первоначального действия .....	174
5.11.3. Фильтрация входного шума .....	174
5.11.4. Предотвращения выхода управляющего параметра за допустимые пределы .....	175
5.11.5. Предотвращение выхода управляющего параметра за пределы средней величины .....	176
5.11.6. Реализация данных примеров .....	177
5.12. В рабочую тетрадь разработчика .....	178
<b>Глава 6. Соленоиды, реле и другие исполнительные устройства</b> .....	179
6.1. Соленоиды .....	179
6.1.1. Реле .....	179
6.1.2. Интерфейс соленоидов и реле .....	180
6.1.3. Включение/удержание реле .....	182
6.2. Нагревательные элементы .....	185

6.2.1. Обрыв нагревателя . . . . .	186
6.2.2. Обрыв датчика . . . . .	187
6.2.3. Нагреватель на основе термометра сопротивления . . . . .	188
6.3. Охладители . . . . .	191
6.3.1. Охладитель Пельтье . . . . .	191
6.3.2. Вентиляторы . . . . .	191
6.4. Светодиоды . . . . .	193
6.4.1. Оптроны с закрытыми оптическими каналами . . . . .	195
6.4.2. Одновременное включение нескольких светодиодов . . . . .	196
6.5. ЦАП . . . . .	198
6.6. Цифровые потенциометры . . . . .	199
6.7. Аналоговые ключи . . . . .	202
6.7.1. Типы аналоговых ключей . . . . .	204
6.7.2. Применение аналоговых ключей . . . . .	204
6.7.3. Мультиплексоры . . . . .	205
<b>Глава 7. Двигатели . . . . .</b>	<b>206</b>
7.1. Шаговые двигатели . . . . .	206
7.1.1. Биполярные и однополярные шаговые двигатели . . . . .	208
7.1.2. Резонанс . . . . .	210
7.1.3. Полушаговый режим двигателя . . . . .	211
7.1.4. Микрошаговый режим двигателя . . . . .	214
7.1.5. Управление шаговым двигателем . . . . .	215
7.1.6. Сквозной ток . . . . .	216
7.1.7. Контроль тока . . . . .	220
7.1.8. ИС управления двигателем . . . . .	221
7.1.9. Управление по методу прерывания . . . . .	222
7.1.10. Методы управления и резонанс . . . . .	224
7.1.11. Линейное управление . . . . .	225
7.2. Двигатели постоянного тока . . . . .	226
7.2.1. Управление двигателями постоянного тока . . . . .	227
7.2.2. Бесколлекторные двигатели постоянного тока . . . . .	230
7.2.3. Энкодеры . . . . .	234
7.2.4. ИС контроллеров двигателей постоянного тока . . . . .	237
7.2.5. Программируемые контроллеры . . . . .	242
7.3. Достоинства и недостатки различных типов двигателей . . . . .	244
7.4. Порядок включения питания логических цепей и двигателя . . . . .	246
7.5. Вращающий момент двигателя . . . . .	248
7.5.1. Вращающий момент при заторможенном двигателе . . . . .	248
7.5.2. Противодействующее электромагнитное поле . . . . .	248
7.5.3. Вращающий момент и скорость . . . . .	248
7.6. Реальное применение шагового двигателя . . . . .	249

---

<b>Глава 8. Электромагнитные помехи</b> .....	254
8.1. Связь по земле .....	254
8.1.1. Ток двигателя .....	258
8.1.2. Самонаведенные токовые ошибки .....	259
8.2. Электростатический разряд .....	260
8.2.1. Самонаведенный электростатический разряд .....	260
8.2.2. Защита от электростатического разряда .....	261
<b>Глава 9. Системы высокой точности</b> .....	264
9.1. Общие сведения .....	264
9.2. Входное напряжение смещения .....	266
9.3. Входное сопротивление .....	267
9.4. Частотные характеристики .....	268
9.5. Влияние температуры на сопротивление резисторов .....	269
9.6. Источник опорного напряжения .....	270
9.7. Влияние изменения температуры в целом .....	272
9.8. Заземление и шумы .....	273
9.9. Печатная плата .....	275
9.9.1. Заземление на печатной плате .....	275
9.9.2. Источник питания .....	277
9.10. Статистическая погрешность .....	278
9.11. Опорное напряжение от источника питания .....	280
9.12. Резюме .....	280
<b>Глава 10. Стандартные интерфейсы</b> .....	281
10.1. Интерфейс IEEE 1451.2 .....	281
10.1.1. Электрические параметры .....	281
10.1.2. Параметры датчиков .....	282
10.1.3. Международная система единиц измерения СИ .....	282
10.2. Последовательный периферийный интерфейс токовая петля 4-20 мА .....	283
10.3. Интерфейс Fieldbus .....	283
<b>Глава 11. Аналоговая схемотехника</b> .....	286
11.1. Питание микроконтроллера и напряжение смещения .....	286
11.2. Наборы резисторов .....	288
11.3. Многоходовые системы управления .....	289
11.4. Управление цепями переменного тока .....	291
11.5. Схемы контроля напряжения питания .....	293
11.6. Управление биполярными транзисторами .....	294
11.6.1. Преобразование логических уровней .....	296
11.6.2. Скорость переключения .....	296
11.6.3. Высоковольтные ключи .....	297
11.7. МОП-транзисторы .....	297

---

11.7.1. Управление МОП-транзисторами . . . . .	297
11.7.2. МОП-транзистор для коммутации . . . . .	299
11.8. Измерение параметров сигнала отрицательной полярности . . . . .	301
11.9. Пример системы управления . . . . .	302
11.9.1. Релейное управление . . . . .	305
11.9.2. Пропорциональное управление . . . . .	305
11.9.3. ПИД-регулирование. . . . .	309
11.9.4. Пропорционально-интегральное управление . . . . .	313
<i>Приложение А. Операционные усилители . . . . .</i>	<i>314</i>
<i>Приложение Б. Широтно-импульсная модуляция . . . . .</i>	<i>326</i>
<i>Приложение В. Полезные сайты в Интернете . . . . .</i>	<i>339</i>
<i>Приложение Г. Программа Python для главы 11. Данные Excel для главы 4 . . . . .</i>	<i>340</i>
Словарь терминов . . . . .	345
Предметный указатель . . . . .	348



## Принятые сокращения

ISR (Interrupt Service Routine) — подпрограмма обработки прерываний.  
NTC (Negative Temperature Coefficient) — отрицательный температурный коэффициент.

АЦП — аналогово-цифровой преобразователь.

ДКВ — двойная коррелированная выборка.

ИОН — источник опорного напряжения (reference).

ИС — интегральная схема.

КД — константа диссипации (рассеивание мощности).

КМОП — комплементарная МОП-структура.

КПТ — коэффициент передачи по току (Current Transfer Ratio, CTR).

МК — микроконтроллер.

МОП — металл-окисел-полупроводник.

МП — микропроцессор.

ОЗУ — оперативное запоминающее устройство (Random Access Memory, RAM).

ОК — открытый коллектор.

ООС — отрицательная обратная связь.

ОС — обратная связь.

ОУ — операционный усилитель.

ПЗС — прибор с зарядовой связью.

ПЗУ — постоянное запоминающее устройство (Read Only Memory, ROM).

ПИД — пропорционально-интегрально-дифференциальный.

ПНЧ — преобразователь напряжение-частота.

ПО — программное обеспечение.

ППЗУ — программируемое ПЗУ.

РТД — резистивный температурный датчик.

УВХ — устройство выборки-хранения.

ФНЧ — фильтр нижних частот.

ЦАП — цифро-аналоговый преобразователь.

ЦОС — цифровая обработка сигналов.

ШИМ — широтно-импульсная модуляция.

ЭДС — электродвижущая сила.

ЭМС — электромагнитная совместимость.

ЭСППЗУ — электрически стираемое ППЗУ.

## Предисловие

Иногда создается впечатление, что между аналоговой и цифровой схемотехникой лежит огромная пропасть. Разработчики цифровой техники не любят погружаться в дебри аналогового мира, а разработчики аналоговых схем — связываться с цифровыми системами. Отличаются даже термины для обозначения одних и тех же понятий.

Несмотря на то что скорость и производительность микропроцессоров стремительно возрастает, а с ней возрастает «цифровизация» аппаратуры различного назначения, окружающий мир остается аналоговым по своей природе. Для измерения процессов в окружающем мире и управления различными устройствами, разработчикам микропроцессорных систем, так или иначе, приходится применять аналоговую технику. Причем существует техническая литература отдельно по аналоговой схемотехнике и отдельно по цифровой. В данной книге предпринята попытка соединить эти разделы радиоэлектроники.

## Введение

В основном современные электронные системы — микропроцессоры, логические устройства, интерфейсы — являются цифровыми. Цифровые вычислительные системы в настоящее время проще создать, чем аналоговые с теми же функциями. К примеру, представьте современный аналоговый персональный компьютер, который должен управлять работой многих программ, выполнять сложные вычисления, обеспечивать выход в Интернет... Хотя на заре компьютерной эры встречались и аналоговые вычислительные машины.

Сегодня электронный мир, в основном, цифровой, реальный же мир и цифровой, и аналоговый. Температура в вашем офисе не просто горячая или холодная, а изменяется в широком диапазоне. Вы можете использовать термометр для определения температуры, но как преобразовать эту величину в цифровой вид для передачи информации в микропроцессор? Микроконтроллер управления зажиганием в автомобиле должен измерять скорость вращения вала двигателя для формирования искры в нужное время. Микропроцессорная система управления станка должна точно позиционировать резец для обработки детали.

В этой книге описаны многие практические методы управления внешними устройствами и некоторые схемы на операционных усилителях (ОУ). Но главной задачей книги не является ни описание теории управления, ни описание теории работы операционных усилителей. В основном, книга посвящена измерению аналоговых величин и управлению аналоговыми устройствами с помощью микропроцессорных систем. Измеряете ли вы сигнал со спутника или температуру в тостере, микропроцессорная система должна измерять, анализировать и управлять аналоговыми величинами. В двух словах можно сказать, что эта книга о соединении входных и выходных аналоговых устройств с микропроцессорной системой для выполнения прикладных задач.

Большинство встроенных микропроцессорных систем содержат узлы обработки входных сигналов и формирования выходных сигналов. При этом как входные, так и выходные сигналы могут быть аналоговыми. Цифровые узлы аналоговых систем, такие, например, как интерфейс микропроцессора (МП) с памятью, в данной книге не представлены. Однако в этой книге рассматриваются компоненты, присущие любой встроенной системе и осуществляющие связь этой системы с внешним миром.

## 1.1. Динамический диапазон

Прежде чем создавать систему, необходимо уточнить требуемый *динамический диапазон входных и выходных сигналов*. Динамический диапазон определяет, с какой точностью должны быть измерены входные сигналы или сформированы выходные. Точность измерения, в свою очередь, влияет на такие параметры, как допустимый уровень шума и точность применяемых компонентов. Допустим, что микропроцессорная система измеряет входной аналоговый сигнал и преобразует его в цифровой код (как это делается, будет показано в главе 2). Динамический диапазон традиционно выражается в децибелах (дБ), как принято при измерении отношений напряжения или мощности. Проще говоря, динамический диапазон можно представить как отношение наибольшей величины к наименьшей. Но этих отношений недостаточно, если речь идет о микропроцессорной системе. Так, в большинстве случаев следует также знать число бит для обеспечения требуемой точности при измерении или управлении.

Так, например, если температура должна измеряться в диапазоне  $0...100^{\circ}\text{C}$  и точность измерения должна составлять  $1^{\circ}\text{C}$ , понадобится 100 дискретных величин. Поскольку 8-битный аналого-цифровой преобразователь (АЦП) может разбить величину входного напряжения на  $2^8$ , или 256, значений, для данной системы вполне допустимо применение 8-бит-

ного АЦП. Но если точность измерения температуры повышается и должна составлять  $0.1^{\circ}\text{C}$ , понадобится уже  $100/0.1$  дискретных величин, т. е. 1000 величин, и, следовательно, — 10-битный АЦП (который производит  $2^{10}$ , или 1024 величины).

## 1.2. Точность измерения напряжения

Число бит, необходимое для измерения температуры в нашем случае, зависит от диапазона измеряемой величины (температуры, напряжения, интенсивности света, давления и т. п.), а не от диапазона входного напряжения АЦП. Действительно, диапазон температур  $0\dots 100^{\circ}\text{C}$  может быть приведен к разным величинам размаха входного сигнала:  $0\dots 5\text{ В}$  или  $0\dots 1\text{ В}$ . В обоих случаях динамический диапазон остается тем же. Однако в диапазоне  $0\dots 5\text{ В}$  для обеспечения точности  $1^{\circ}\text{C}$  каждый шаг составляет  $19.5\text{ мВ}$ , который получается делением  $5\text{ В}/256$ , а для обеспечения точности  $0.1^{\circ}\text{C}$  —  $4.8\text{ мВ}$  ( $5\text{ В}/1024$ ). В случае диапазона  $0\dots 1\text{ В}$  каждый шаг уменьшается соответственно до  $3.9\text{ мВ}$  и  $976\text{ мкВ}$ . Это повлияет на выбор типов АЦП и операционных усилителей (ОУ) и т. п. (Более детальное изучение этих соотношений будет произведено в последующих разделах.) При этом динамический диапазон системы определяет, сколько бит необходимо для измерения с требуемой точностью величин сигналов. То, как преобразуется динамический диапазон аналогового и затем цифрового сигнала в целом, накладывает определенные ограничения на систему.

## 1.3. Калибровка

Определение параметров, связанных с динамическим диапазоном, поднимает вопросы калибровки. Измеряемая величина, заключенная в определенном динамическом диапазоне, подразумевает определенное число бит для достижения требуемой точности. Однако реальные компоненты, применяемые для измерения реальных величин, сами обладают собственными погрешностями. Резистор номиналом  $10\text{ кОм}$  может иметь сопротивление в пределах  $9900\dots 10100\text{ Ом}$ , если погрешность составляет  $1\%$ , или —  $9990\dots 10010\text{ Ом}$ , если погрешность составляет всего лишь  $0.1\%$ . К тому же, сопротивление изменяется при изменении температуры. Все компоненты системы, включая датчики, имеют погрешности. Более детально этот вопрос будет изучен в главе 9, а сейчас рассмотрим, как достигается требуемая точность.

Возьмем вновь пример с измерением температуры в диапазоне  $0\dots 100^{\circ}\text{C}$ . Измерение с точностью  $1^{\circ}\text{C}$  может быть достигнуто без дополни-

тельных подстроек. Однако надо заметить, что точность  $0.1^{\circ}\text{C}$  потребует введения дополнительной калибровки, поскольку трудно найти соответствующий температурный датчик по приемлемой цене. Выйти из положения можно, включив в конструкцию подстроечный элемент для компенсации погрешности.

Необходимость этапа калибровки подразумевает и другие технические требования к системе. Например, будет ли часть системы, содержащая датчик, находиться на плате рядом с подстроечным элементом? Если нет, то, как связать эти компоненты после того, как калибровка выполнена? И что, если инженер должен заменить датчик в производственных условиях? Сделает ли инженер калибровку? Будет ли реально дешевле добавлять этап калибровки при процедуре сборки, чем купить более дорогой датчик? На данном этапе технические и экономические вопросы калибровки идут рядом и должны учитываться разработчиком. Решить все эти вопросы и сделать калибровку технологичной, независимой от образца к образцу поможет замена аналоговых высокоточных элементов и датчиков на стандартные цифровые компоненты.

Например, во многих случаях, когда без калибровки не обойтись, полученные параметры могут быть вычислены программно и записаны в память системы. Как этого добиться? Можно поместить датчик в среду с известной температурой и измерять выходной сигнал. Известно, что идеальный датчик должен произвести напряжение  $X$  при температуре  $T$ , однако реальный датчик при температуре  $T$  производит напряжение  $Y$ . Измеряя выходной сигнал при различных температурах, можно получить таблицу соответствия напряжение/температура в виде констант. Данная таблица констант может быть сохранена в памяти микропроцессорной системы. Когда микропроцессор будет считывать показания датчика, он обратится к данной таблице констант и произведет необходимые вычисления для определения действительной температуры.

Допустим, вам понадобилось хранить калибровочные константы датчика, который физически размещается на некотором удалении от микропроцессора. На **Рис. 1.1** показаны три варианта хранения калибровочных констант. На **Рис. 1.1а** удаленный датчик соединен кабелем с микропроцессором. Тогда датчик можно было бы заменить другим без дополнительной калибровки, так как микропроцессор хранит калибровочные данные во внешнем электрически стираемом ППЗУ (ЭСППЗУ — Electrically Erasable Programmable Read-Only Memory, EEPROM) или флэш-памяти (flash memory). Особенности использования данной схемы подключения следующие:

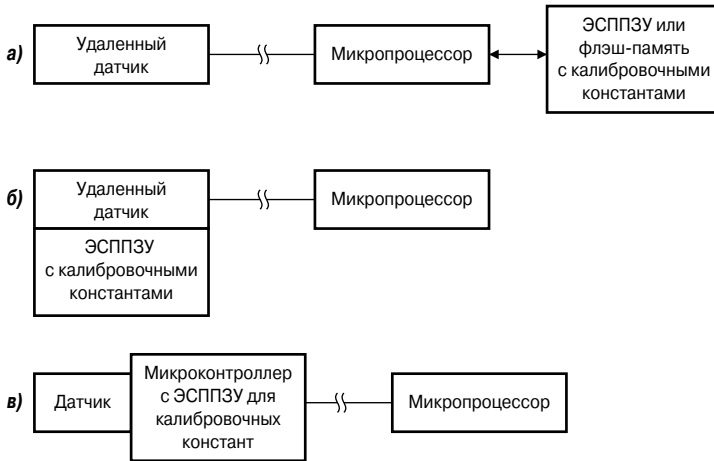


Рис. 1.1. Методы хранения калибровочных констант датчика

- Калибровочные константы для всех используемых датчиков можно хранить в одной и той же области памяти, что требует меньшего объема памяти. К тому же, если калибровка выполняется с помощью программных вычислений, вместо поиска из таблицы констант, одна подпрограмма может быть использована для любых датчиков одного типа.
- После калибровки датчик должен всегда находиться с одной и той же микропроцессорной платой.
- Если заменить датчик или микропроцессор, используя калибровочные константы одного датчика для другого, то результаты будут неточными, но узнать об этом будет невозможно, пока микропроцессор не идентифицирует новый датчик.

Схема на **Рис. 1.1б** иллюстрирует другой метод хранения констант совместно с микропроцессором, а в ППЗУ, содержащем калибровочные константы, которое находится на одной плате с датчиком. ППЗУ может представлять собой небольшую микросхему, соединенную с микропроцессором при помощи одного из типов интерфейса: I<sup>2</sup>C или Microwire (более подробно об интерфейсах см. в главе 2). Особенности данной схемы подключения следующие:

- Поскольку каждый датчик снабжен ППЗУ со своими калибровочными константами, плата датчика может быть заменена без дополнительных трудностей. Датчики могут быть откалиброваны без специального согласования с микропроцессорной системой.

- Требуется больше запоминающих устройств — по одной интегральной схеме ППЗУ на каждый датчик.

И, наконец, схема на **Рис. 1.1в** находится еще на шаг впереди своих предшественников. На плату с датчиком устанавливается микроконтроллер (МК), выполняющий калибровку и хранение данных во внутреннем ППЗУ микроконтроллера.

Особенности третьей схемы следующие:

- Отпадает необходимость в калибровке на плате главного микропроцессора. При одинаковых измеряемых величинах на центральную плату всегда будет поступать одна и та же величина независимо от разброса параметров датчиков.
- Если датчик недоступен и должен быть заменен в готовом изделии, замена не повлияет на программное обеспечение главного процессора, поскольку все калибровки сохранены на удаленной плате во внутренней памяти микропроцессора датчика.
- Требуется поддерживать большее количество микропроцессорной техники, в некоторых случаях, с серьезным программным обеспечением (медицина, военная техника), что может значительно увеличить стоимость.

Отдельная тема при рассмотрении процесса калибровки — это человеческий фактор. Если система нуждается в калибровке в условиях производства, не окажется ли так, что персонал должен выполнять калибровку на расстоянии около 3.66 м (12 футов) от центрального процессора и одновременно нажимать клавишу ENTER на клавиатуре? Где же следует разместить переключатели так, чтобы не делать постоянные пробежки от датчика к компьютеру для нажатия клавиш или просмотра результатов на дисплее? Следует ли автоматизировать процесс настройки для снижения числа действий оператора? Чем больше ручных подстроек, тем больше шансов ошибиться.

## 1.4. Пропускная способность шины

Несколько лет назад я занимался методами визуализации изображений. Разработанная система служила для ввода данных с помощью светочувствительной ПЗС-линейки. За одно сканирование мы регистрировали 1024 *пикселя*<sup>1)</sup>. К тому же, мы должны были считывать информацию со скоростью 150 дюймов в секунду с разрешением 200 пикселей на дюйм. Сигнал от каждого элемента подвергался 8-битному аналого-цифровому

<sup>1)</sup> Пиксель (от англ. picture element — pixel) — точка растра. (*Прим. науч. ред.*)



преобразованию (байт на пиксель). Скорость обработки данных составила, таким образом,  $150 \times 1024 \times 200$  или 30 720 000 байт/с.

Планировалось использовать шину *стандартного параллельного интерфейса магистрально-модульного VME* (Versa Module Europe bus) как основную для данной системы. Данные каждого сканирования должны были быть считаны, нормированы, отфильтрованы, а затем преобразованы в 1-битное изображение. При обсуждении архитектуры будущей системы один из инженеров настаивал, чтобы все данные были поданы на шину VME. В те дни шина VME характеризовалась максимальной пропускной способностью (bandwidth) 40 Мбайт/с, но очень немногие системы достигают максимально возможного теоретического предела. Нам была необходима следующая пропускная способность:

- Считывание данных, поступающих от видеокамеры: 30.72 Мбайт/с.
- Прохождение данных через систему нормализации: 30.72 Мбайт/с.
- Прохождение данных через фильтр: 30.72 Мбайт/с.
- Поступление данных на преобразователь в монохромный формат: 30.72 Мбайт/с.
- Прохождение монохромных данных к выходу: 3.84 Мбайт/с.

Если сложить данные значения, получится величина 126.72 Мбайт/с, намного превышающая даже теоретический предел пропускной способности шины VME. Недавно мне довелось работать с подобным устройством визуализации изображений с применением систем *цифровой обработки сигналов*, ЦОС (Digital Signal Processing, DSP) и множественной системной 32-битной шиной *стандартного параллельного периферийного интерфейса* для подключения периферийных устройств *PCI* (Peripheral Component Interconnect), причем одна из шин PCI была близка к своей максимальной пропускной способности.

Основная задача в таких случаях состоит в том, чтобы знать, какой объем данных требуется передать, и какие используются интерфейсы для передачи данных. Если предполагается использовать *стандартные последовательные интерфейсы*, такие как *Ethernet* или *FireWire*, следует удостовериться в том, что они поддерживают требуемую пропускную способность.

## 1.5. Производительность процессора

Во многих приложениях необходимо учитывать *производительность процессора* (Processor Throughput). В упомянутых выше системах обработки изображений большинство функций было реализовано аппаратно, поскольку существующие на тот период времени процессоры не могли обеспечить достаточной скорости. С ростом тактовой частоты процессоров

больше функций стало выполняться программно. Существуют отдельные ключевые факторы, которые необходимо рассмотреть для определения производительности процессора.

### 1.5.1. Прерывания

Как часто должны происходить прерывания, и как много операций должно быть выполнено в каждой процедуре обработки прерывания (Interrupt Service Routine, ISR)? Какова максимальная задержка обработки прерывания? Будут ли некоторые прерывания запрещены на продолжительное время, и как это отразится на задержке остальных прерываний? Вам может понадобиться один или более процессоров: один для выполнения высокоскоростных прерываний с минимальным временем задержки и невысокой сложностью вычислений, другой для обработки медленных прерываний с повышенными требованиями к программному обеспечению.

### 1.5.2. Интерфейсы

Что система должна сообщать? Как будет осуществляться связь с внешним миром? Сколько потребуется электроники для поддержания выбранного интерфейса, и как много функций должно будет выполнять программное обеспечение? Например, интерфейс I<sup>2</sup>C, осуществляемый программным обеспечением микроконтроллера, больше повлияет на общую производительность, чем тот же интерфейс, но выполненный аппаратно. Подобная ситуация возникает и в связи с обработкой прерываний, поскольку в интерфейсах также используются прерывания. Об интерфейсе I<sup>2</sup>C более подробно рассказывается в главе 2.

### 1.5.3. Поддержка на аппаратном уровне

Система визуализации изображений, в которой используется контроллер прямого доступа к памяти, ПДП (Direct Memory Access, DMA), не потребует столько вычислительной мощности от процессора, как система, в которой все данные обрабатываются программно. Процессор, передающий данные программе, но обладающий аппаратной инструкцией перемещения блока (block-move инструкцией), будет быстрее процессора, выполняющего цикл за циклом. Подобно тому, как, если микропроцессор содержит встроенный математический сопроцессор, то операции с плавающей точкой выполняются аппаратно быстрее, чем с помощью программы. В каждом конкретном случае необходимо выбрать оптимальное соотношение программных и аппаратных средств в зависимости от решаемой задачи и расставленных приоритетов.

#### 1.5.4. Требования к процессору

При обработке изображений передача процессором данных от одного блока (например, интерфейс камеры) к другому (например, цифровой фильтр) отнимет некоторую часть процессорного времени. Если процессор должен выполнить алгоритм фильтрации программно, это займет значительную часть процессорной мощности. Неправильно спроектированные процессорные системы, не учитывающие объем операций микропроцессора, как правило, выполняют операции медленно и круг их задач крайне ограничен.

#### 1.5.5. Требования к операционной системе

При использовании операционных систем следует обратить особое внимание на время запрета выполнения прерываний. Совместимо ли это время с ограничениями по задержке на обработку прерываний? А если операционная система вдруг остановит вычисления для того, чтобы потратить несколько секунд на обслуживание жесткого диска? Станет ли это причиной потери данных? Должна ли система работать в режиме реального времени (что потребует применения специальной операционной системы реального времени)?

#### 1.5.6. Язык программирования и компилятор

Если предполагается использовать *объектно-ориентированный язык программирования высокого уровня*, такой как C++, то что может случиться, если процессору понадобится переслать большой объем данных в память? Не потеряется ли часть данных? Не придется ли менять 100 МГц процессор на 1 ГГц, только для того, чтобы сократить интервал времени сбора данных?

### 1.6. Ограничение скорости

Выбор достаточно быстрых шин и процессора — очень важное дело, но может не менее важно избежать слишком высокой скорости? Может показаться нелогичным нежелание применять самый быстрый процессор и периферийные устройства, однако бывают приложения, где встречается именно такая ситуация, не требующая суперпроцессора... В основном, из-за цены или в соответствии с требованиями электромагнитной совместимости устройств.

### 1.6.1. Затраты

*Стандартный компьютерный параллельный интерфейс PC/104* определяет механические и электрические характеристики плат PC, оптимизированных для встроенных систем. Платы PC/104 поставляются с оригинальной шиной, подобной устаревшей шине *стандартного компьютерного параллельного интерфейса ISA* по электрическим и временным характеристикам, но способной передавать данные со скоростью 5 Мбайт/с. Платы центрального процессорного устройства (ЦПУ) компьютера работают на повышенных скоростях обмена информацией при использовании усовершенствованного *стандартного компьютерного параллельного интерфейса PC/104+*. При этом скорость обмена доходит до скоростей по интерфейсу PCI (133 Мбайт/с). Хотя часто предпочтение отдается более скоростным устройствам, следует, однако, учитывать, что цена платы PC/104 значительно ниже, чем PC/104+. Благодаря синхронизации на меньшей частоте, PC/104 позволяет применять более длинные проводники и более простую логику. Если перед вами стоит задача подключения большой (по размерам) платы аналогового ввода/вывода к плате PC/104, то временные соотношения PC/104 позволят вам сделать проще трассировку соединений. С другой стороны, при выпуске систем малым тиражом, может не получиться покрыть все расходы на применение скоростной платы PC/104+. Подобные рассуждения применимы, в общем случае, и для построения систем на основе других компьютерных архитектур типа PC, Compact PC и т. п. Все зависит от того, что мы хотим и за какие деньги.

### 1.6.2. Электромагнитная совместимость

Почти каждая микропроцессорная система проходит тест на *электромагнитную совместимость (ЭМС)* перед тем, как поступить в продажу в США или в Европе. Требования ЭМС ограничивают объем энергии, излучаемой прибором, для предотвращения влияния на другие электроприборы, такие как телевизоры или радиоприемники. Заметим, что чем выше тактовая частота, тем сильнее будет излучение в окружающее пространство. Современные стандарты ЭМС охватывают электромагнитное излучение в частотном диапазоне 30 МГц...1 ГГц. Процессор, работающий на частоте 6 МГц, не будет обладать значительной излучающей способностью в том диапазоне, в который могут попасть лишь пятая и выше гармоники тактовой частоты процессора. Высшие гармоники обычно обладают меньшей энергией. Напротив, 33-МГц процессор генерирует излучение в тестируемом диапазоне уже на основной частоте. К тому же, микропроцессор, работающий на

повышенной частоте, требует применения скоростных логических ИС для обработки сигналов с короткими фронтами и соответственно увеличенной энергией высших гармоник. Пример с 6-МГц процессором в эпоху 3-ГГц 2-ядерных Пентиумов слишком архаичен, но вносит большую ясность. Ограничения по ЭМС — зачастую именно это причина снижения скорости процессора и шины до частот, действительно необходимых для работы. Естественное предупреждение в данном случае — не ограничивать частоты слишком, чтобы не оказаться на грани работоспособности прибора.

## 1.7. Другие системные ограничения

### 1.7.1. Периферийные устройства

Однажды, у меня при разработке системы визуализации возникли проблемы с потерей данных. Дело в том, что система временно записывала в буфер жесткого диска значительный объем данных. Проблема крылась в функции записи жесткого диска; диск ненадолго останавливался при приеме данных, и буфер с файлом изображения переполнялся. Этот дефект связан с тем, что используемый жесткий диск оснащен схемой температурной компенсации, которая часто требовала от встроенной схемы управления отключения обмена данными на несколько десятков миллисекунд. Наша система, однако, требовала довольно продолжительного доступа к диску. Так что удостоверьтесь сначала, что используемые вами *периферийные устройства* не создают лишних проблем.

### 1.7.2. Общие интерфейсы

Что влияет на общие интерфейсы? Например, если записывать с двух видеокамер данные на два жестких диска, то скорости одного интерфейса IDE может не хватить. Может понадобиться два интерфейса IDE для двух дисков так, чтобы те могли работать независимо либо придется выбрать другой интерфейс жесткого диска. Удержит ли все данные 10-baseT Ethernet или сразу выбрать 100-baseT Ethernet? Сопоставьте объем данных с возможностями выбранного интерфейса и удостоверьтесь в том, что он обладает необходимой пропускной способностью.

### 1.7.3. Приоритет задач

Архитектура IBM PC используется для многих приложений. Это хорошо документированный стандарт с огромным числом совместимого программного обеспечения, имеющегося в наличии, от разных производителей.

лей. Но существуют и недостатки, включая работу *операционной системы Windows* не в режиме реального времени. Вы, вероятно, сталкивались с временным прекращением на несколько секунд реакции компьютера на внешние воздействия из-за переполнения емкости жесткого диска по непонятным причинам. При печати документа, например, некоторое раздражение вызывает такое явление, когда курсор вдруг пропадает и появляется через несколько секунд вместе с набором вновь введенных символов. Что произойдет, если операционная система ПК вдруг прекратит интенсивный поток данных от аудио- или видеоустройства к ПК, работающего в качестве домашнего кинотеатра, и начнет выполнять другую задачу? Если в компьютерной системе не предусмотрено предоставление наивысшего приоритета вашей задаче, то неизбежна потеря данных. При использовании РС-подобной архитектуры следует удостовериться, что выбранные вами программное обеспечение и операционная система будут поддерживать необходимые функции.

#### 1.7.4. Системные требования

Нужен ли вам процессор, способный проводить вычисления с плавающей точкой для обработки данных? Если так, то простым 8-битным микроконтроллером не обойтись. Понадобится, по крайней мере, компьютер на основе 486-го процессора. Потребуется ли для выбранной скорости передачи данных процессор с прямым доступом к памяти? Подобное изучение круга решаемых задач помогает выбрать подходящий микропроцессор. В некоторых случаях, возможно, лучше немного переделать технические средства, работающие совместно с микропроцессором для снижения цены прибора, чем применять дорогостоящий экземпляр процессора. Данное утверждение иллюстрируется на следующем примере.

Представьте, что колесо, приводимое в движение двигателем, производит прерывание в вашей программе через каждые 20 градусов поворота (**Рис. 1.2**). Двигатель может вращаться с разной скоростью, и процессор должен отслеживать эти изменения, так как включает соленоид, чтобы открыть клапан через несколько градусов вращения после вызова процедуры обработки прерывания.

Такие 20-градусные прерывания будут повторяться через каждые 3.3 мс, если колесо вращается со скоростью 1000 оборотов в минуту, либо через каждые 666 мкс, если скорость вращения равна 5000 об./мин. Если процессор использует таймер для измерения скорости вращения (времени между прерываниями) с тактовой частотой 1 МГц, то между прерываниями на скорости 1000 об./мин таймер сделает 3300 отсчетов, а на скорости 5000 об./мин — 666 отсчетов.

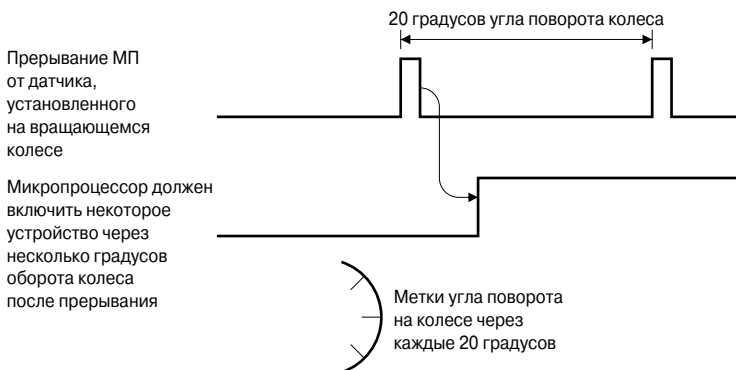


Рис. 1.2. Временные диаграммы вращения колеса

Скажем, процессор должен включить соленоид после поворота колеса на 5 градусов после очередного прерывания, как показано на **Рис. 1.2**. Формула для вычисления времени таймера (сколько должно быть добавлено к текущему отсчету для задержки на 5°) выглядит следующим образом:

$$\text{Приращение таймера} = \frac{5^\circ \text{ задержка}}{20^\circ / \text{прерывание}} \times N,$$

где  $N$  — число отсчетов таймера на прерывания.

Так, на скорости 1000 об./мин задержка 5° составит 825 отсчетов таймера, а на 5000 об./мин — 166 отсчетов. Применение данного метода во встроенных системах с использованием ограниченных по производительности процессоров проблемно из-за необходимости проведения операции деления. Деление очень затратная по времени операция для программного обеспечения, и для данного метода лучше выбрать процессор со встроенной инструкцией деления.

Если заменить нашу измерительную систему так, чтобы двадцатиградусные метки были поделены на двоичные величины, вычисления становятся гораздо проще. Скажем, мы решили поделить 20-градусные метки на 32 равные части, каждая — по 0.625°. Теперь переименуем наши приращения. Приращение 5° будет теперь 5/0.625 или 8 единиц. Формула будет выглядеть следующим образом:

$$\text{Приращение таймера} = \frac{8}{32 / \text{прерывание}} \times N.$$

Это даст нам тот же результат, что и ранее (825 на скорости 1000 об./мин и 166 на скорости 5000 об./мин), но деление на 32 может быть выполнено с помощью простой операции сдвига в сдвиговом регистре, вместо сложного программного алгоритма. Мы, путем дополнительной разметки, перешли от десятичной к двоичной системе счисления, в которой выполняет вычисления микропроцессор. Подобное изменение позволяет применять простой 8-битный микроконтроллер вместо сложного и дорогого микропроцессора. Все, что мы сделали — это перешли от измерения скорости вращения к измерению того, что легче будет считать процессору. Это пример нетривиального выхода из положения, если не по карману покупка дорогого микропроцессора.

### 1.7.5. Разрядность цифрового слова

Если перед вами стоит задача соединения 12-битного АЦП с микропроцессором, вы, вероятно, предпочли бы 16-битный процессор вместо 8-битного. Хотя 16-битные операции можно исполнить и на 8-битном микропроцессоре, но это требует обычно множества инструкций и накладывает другие ограничения. Если процессор не просто передает данные, а обеспечивает взаимосвязь различных устройств, то необходимо также организовать интерфейс с определенной разрядностью (шириной) слова. Если же производятся вычисления с 32-битной точностью, то более приемлемым покажется 16-битный процессор с шириной слова 32-бита для более быстрых и легких вычислений.

### 1.7.6. Интерфейсы

Удостоверьтесь, что какое-либо из возможных состояний интерфейса не нанесет вред системе. Например, плата микропроцессора соединена с контроллером двигателя при помощи кабеля. Что произойдет, если кто-либо из обслуживающего персонала оставит кабель неподключенным и включит систему? Останется двигатель в первоначальном положении или начнет двигаться бесконтрольно? Необходимо учесть все подобные варианты.

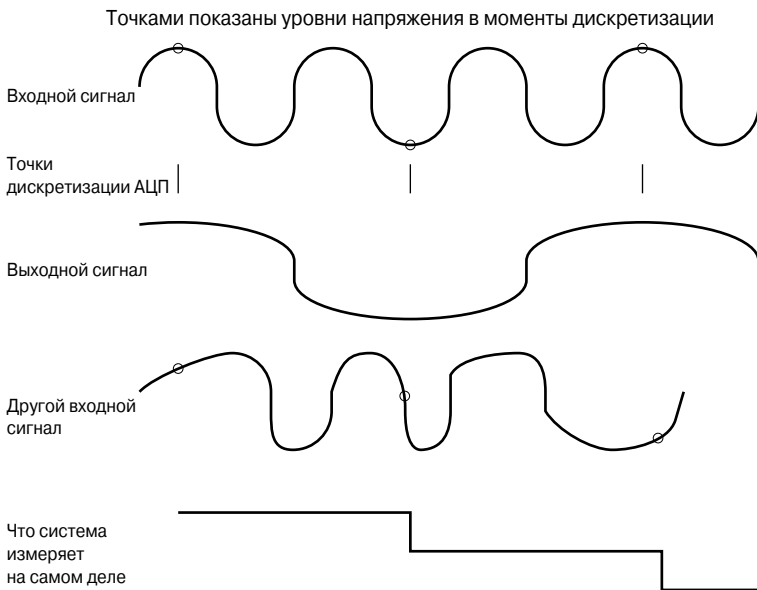
## 1.8. Частота дискретизации и наложение

На **Рис. 1.3** показан входной синусоидальный сигнал АЦП и выходной сигнал при недостаточно высокой частоте дискретизации. При аналого-цифровом преобразовании синусоиды заданной частоты с вдвое меньшей частотой дискретизации, АЦП измеряет кажущуюся синусоиду вдвое



меньшей частоты. Этот эффект называется наложением спектров (aliasing) и происходит всякий раз, когда частота входного сигнала кратна высокой частоте дискретизации.

Также на **Рис. 1.3** показан входной сигнал другой формы — не синусоидальный. В этом случае сигнал, сохраняемый в дискретных отсчетах и вовсе не совпадает с входным.



**Рис. 1.3.** Наложение

Заметим, что любая система должна быть спроектирована так, чтобы не исказить измеряемую величину. Должна быть учтена и скорость, с которой АЦП дискретизирует входной сигнал, и скорость, с которой АЦП может обрабатывать полученные отсчеты. Как известно, каждый непрерывный сигнал можно представить в виде набора выборок (дискретов), период которых, согласно *теореме Шеннона*<sup>1)</sup>, должен быть обратно про-

<sup>1)</sup> В отечественной научной литературе эту теорему называют теоремой Котельникова по фамилии ученого, сформулировавшего эту теорему независимо от иностранного автора. (*Прим. науч. ред.*)