**1. Что такое чип-карта и ее применение для санкционированного доступа к информации**

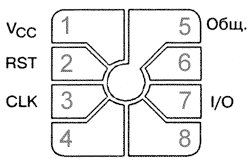
Чип-карта – это пластина из полимерного материала размерами 85×54 мм и толщиной 0,76 мм, с расположенными внутри ее микропроцессором, памятью, и с контактной площадкой, служащей для «общения» карты с терминалом. Чип-карты (в дальнейшем просто карта) используются во многих областях: в банковской сфере, для оплаты разговоров с телефонов-автоматов, для просмотра кодированных каналов с помощью абонентских приемников, в сотовых телефонах (там используется меньшая по размерам карта, названная SIM-картой).

Упрощенно принцип работы карты можно описать так: в памяти карты хранятся некие секретные ключи, и терминал, желая проверить, имеет ли пользователь право доступа, посылает на карту запрос, который обрабатывает микропроцессор карты, выдавая нужный ключ. В случае спутникового терминала, этим ключем производится расшифровка цифрового телевизионного сигнала, так что никакие переделки в самом приемнике не помогут «на халяву» смотреть зашифрованные программы – нужно знать ключи и алгоритм обмена ключами между картой и спутниковым приемником.

*Пример французской телекарты:*

Первоначально контактные площадки карт выполнялись по стандарту AFNOR (контактная площадка была сдвинута к верхнему краю карты), затем был принят стандарт ISO 7816–2 и все совремменые карты выполняются по этому стандарту (как раз такая и показана на рисунке выше).

*Назначение выводов контактной площадки стандарта ISO 7816:*



|  |  |
| --- | --- |
| 1 Vcc | Напряжение питания (+5 В) |
| 2 Reset | Сигнал сброса карточки |
| 3 Clock | Синхронизация |
| 4 Reserved |  |
| 5 Gnd. | Общий |
| 6 Reserved |  |
| 7 I/O | Линия последовательного ввода / вывода данных |
| 8 Reserved |  |

Вообще существуют три разновидности стандарта ISO 7816:

ISO 7816–1 определяет физические характеристики карт

ISO 7816–2 задает размеры и расположение контактов

ISO 7816–3 стандартизирует сигналы и протоколы связи. Карты бывают синхронные и асинхронные. Телефонные карты обычно синхронные, они менее защищены (зачастую там даже нет процессора – только память). В системах шифрования телевизионных сигналов используются асинхронные карты. Любая асинхронная карта согласно стандарту ISO 7816–3 должна выдавать особое сообщение длинной не более чем в 33 байт сразу после посылки комманды Reset. Этот ответ называется ATR (answer to reset) и служит своеобразной «личной» подписью карты (его можно видеть в некоторых программах для работы с картами, но об этом позднее). Самый первый байт имеет особое значение, и должен быть равен 3Fh или 3Bh. Если карта в выдает другое значение – значит это нестандартная либо синхронная, либо дефектная карта. Обмен данными с картой происходит в полудуплексном режиме с помощью контакта номер 7 (I/O).

Для посылки и приема команд с помощью компьютера на карту используется так называемый *connector*. Обычно в качестве connector'а используется **Smartmouse** или **Phoenix**‑интерфейс, который подсоединяется к последовательному (COM) порту компьютера. (Отмечу широко распространенное заблуждение о том, что Smartmouse/Phoenix является программатором. Это не программаторы! Ими нельзя ничего запрограммировать!)

**2. Создание «фальшивой» чип-карты**

«Фальшивую» чип-карту, которая действует так же или почти так же, как и настоящая можно создать, используя микрочипы семейства PIC (обычно используется PIC16F84) компании Microchip, а так же ATMEL AT90S8515 компании Atmel. Чип Atmel гораздо мощнее, чем PIC16F84, однако пиратские карты на его основе (получившие название FunCard) пока еще менее распространены. В этом разделе мы рассмотрим создание карты на микросхеме PIC16F84.

*Краткая характеристика PIC16F84:*

Законченный 10 Mhz CMOS микроконтроллер;

1 К энергонезависимой памяти для программ (по 14 бит);

64 байт энергонезависимой памяти данных;

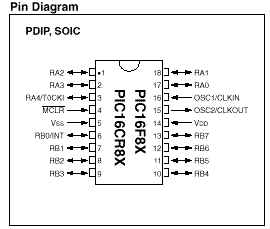
36 регистров общего назначения;

возможность программирования последовательным способом (по любому проводу данных);

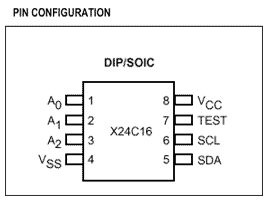
сверхнизкое энергопотребление – 2V – 5V;

возможность подключения дополнительной EEPROM;

стоимость меньше 2.5 USD



Подробнее о чипе 16C84 (у 16F84 с ним много общего) можно прочитать здесь (на русском языке): 16c84.zip.Так как в чипе PIC16F84 очень мало энергонезависимой памяти, в современных пиратских картах для хранения данных дополнительно используют отдельный чип памяти: EEPROM 24C16, емкостью 16 килобит или 2 килобайта (2048x8). Стоит 24C16 недорого (~0.5 USD).



Обмен данными с чипом памяти происходит по протоколу I2C, который состоит из двух линий (не считая общего провода Vss и питания Vcc): SCL (синхронизация) и SDA (данные). Протокол I2C поддерживает адресацию устройств, что позволяет подсоединять к шине несколько устройств разного типа. Более подробно о микросхеме 24C16 и о протоколе I2C можно прочитать в этом файле: x24c16.pdf или здесь (на русском языке): I2C.html.

*Карточки* Итак, для просмотра кодированных каналов в кодировке SECA/Mediaguard или IRDETO нужна карточка, содержащая микрочип PIC16F84 и энергонезависимую память EEPROM 24C16 (эти карточки называют двухчиповыми, в отличие от одночиповых, содержащих только PIC). Внешний вид такой карточки представлен на рисунке.

PIC16F84

EEPROM 24C16

В микросхему PIC программатором зашивается микропрограмма, а в EEPROM коды для раскодировки каналов и другая служебная информация. PIC можно прошивать через контактную площадку карточки, а EEPROM для перепрошивки нужно выпаивать из карточки или воспользоваться Smartmouse/Phoenix‑интерфейсом. Т.к. коды приходится менять довольно часто, то удобнее сделать карточку с панельками (в народе их называют «кроватки»), чтобы было легко вынимать микросхему перед перепрошивкой:

Электрическая схема карточки проста:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Контактная площадка |  | PIC16F84 |  | 24C16 |
|  |  |  |  |  |
| **5** (Vss) | – | **5** (Vss) | – | **1,2,3,4,7** (A0, A1, A2, Vss) |
| **1** (Vcc) | – | **14** (Vcc) | – | **8** (Vcc) |
| **2** (RST) | – | **4** (MCLR) |  |  |
| **3** (CLK) | – | **12, 16** (RB6, CLKIN) |  |  |
| **7** (I/O) | – | **13** (RB7) |  |  |
|  |  | **10** (RB4) | – | **5** (SDA) |
|  |  | **11** (RB5) | – | **6** (SCL) |

Печатную схему двухчиповой карточки (для микросхем с формат-фактором корпуса DIP) от Solo можно взять здесь: SoloPicCard.rar. Если же у вас микросхемы в корпусе для поверхностного монтажа (SMD), схему можно взять отсюда smd-piccard.html.

**3. Сборка программатора типа Ludipipo/JDM**

Для программирования чип-карты нужен программатор. Одним из наиболее простых в изготовлении является программатор Solo JDM, являющийся модификацией традиционного программатора Ludipipo. Его электрическую схему можно взять здесь: JDM\_Scheme.gif. Этим программатором можно программировать микросхемы PIC и EEPROM, просто вставив их в соответствующие панельки (по очереди, конечно), а так же PIC, впаянный в карту. EEPROM на карте запрограммировать им нельзя, т. к. на контактную площадку карты не выведены контакты Data и Clock от EEPROM'а. Но это отнюдь не значит, что намертво впаянную в карту микросхему памяти нельзя запрограммировать никаким образом. Для этого существует интерфейсы Smartmouse либо Phoenix, о которых пойдет речь в пятой части этой статьи. Полное описание программатора JDM от Solo есть в этом архиве: jdm\_prog.rar. От себя лишь добавлю, что для соединения программатора с компьютером нуль-модемный кабель не подойдет, тем более, что на программаторе впаивается стандартный 9‑контактный RS232 разьем типа «мама». При желании, можно обойтись вообще без кабеля, вставив плату прграмматора прямо в гнездо COM на задней стенке системного блока.

**4. Программирование чип-карт**

Итак, все очень просто:

**A.** Вставляем микросхему PIC в панельку программатора (при этом стараемся вставить ее прямо, а не шиворот-навыворот), запускаем программу IC-Prog (ее, а так же другие программы, упоминаемые в этой статье, можно найти у меня в разделе софт).

Сначала нужно настроить программу, для чего в меню «Settings – Hardware» выбрать тип программатора JDM, и номер COM‑порта, к которому он присоединен. Далее:

1. Выбираем тип микросхемы: «PIC 16F84»

2. Загружаем файл прошивки

3. Выбираем Oscillator «XT»

4. Убираем галочку CP (Code Protect), если мы не хотим защитить записываемую прошивку от считывания.

5. Нажимаем кнопку программирования. После того, как прошел процесс программирования, программа автоматически выполнит верификацию (проверку), что все корректно записалость. Если вы поставили галочку CP, проверка не пройдет, т.к. из микросхемы информацию считать будет невозможно.

Если при прошивке будут возникать ошибки, попробуйте поиграться с настройками. Мне пришлось поставить I/O Delay на максимум (меню «Settings – Hardware») для корректного программирования. Для считывания впоследствии прошивки из pic'а воспользуйтесь кнопкой Read All (6).

**B.** Вынимаем прошитую микросхему PIC и вставляем в соседнюю панельку микросхему EEPROM. Выбираем в настройках прогаммы «24C16» и далее поступаем аналогично предыдущему пункту. Настроек при прошивке памяти даже меньше чем для PIC. Микрочип PIC можно прошивать не вынимая его из карты. Необходимо просто вставить карту в card-reader программатора и далее следовать пункту A.

**5. Сборка интерфейса Smartmouse/Phoenix**

Как уже было сказано выше, микросхему памяти, запаянную в карте, программатором JDM прошить невозможно. Но ведь ее чаще всего и необходимо прошивать, т. к. коды к пакетам каналов хранятся именно в ней. Но выход есть, причем весьма хитрый. Дело в том, что карта, как уже было сказано выше, устроена так, что терминал посылает запросы карте, а микропроцессор карты отвечает на них соответствующим образом. Такой диалог между картой и терминалом происходит везде: в банкоматах, спутниковых приемниках, телефонах-автоматах. Существует соответствующее устройство сопряжения и для компьютера, называется оно Smartmouse/Phoenix. Используя его и соответствующую программу можно посылать запросы карте и получать на них ответы, изучая таким образом карту. Интерфейс Smartmouse/Phoenix может работать с любой асинхронной картой, легальной и нелегальной, сделанной в соответствии со стандартом. Например, с помощью этого устройства можно посылать запросы на обновление ключей в легальных картах (если взломана соответствующая система кодировки, иначе неизвестно, какой запрос посылать). Так вот, можно запрограммировать процессор pic‑карты так, что бы все посылаемые данные на карту процессор пересылал в микросхему EEPROM (ведь у процессора нет никаких проблем с доступом к чипу памяти). Соответствующая прошивка процессора называется загрузчиком (Loader).

Алгоритм прошивки всей карты таков:

Берем программатор JDM и прошиваем в PIC программу-загрузчик (она обычно идет в комплекте с программой WinPhoenix).

С помощью Phoenix‑интерфейса программой WinPhoenix пересылаем на карту прошивку EEPROM'а, которая благополучно (я надеюсь) попадает туда, куда надо, т.е. в микросхему EEPROM.

Снова берем программатор JDM и зашиваем в PIC ту прошивку, с которой должна работать карта.

Взять схему интерфейса SmartMouse/Phoenix, да еще и + Season можно здесь: 3in1.rar (авторская разработка SOLO). Интерфейсы Smartmouse и Phoenix отличаются между собой лишь частотой (у Phoenix 3.5Mhz, у Smartmouse 6Mhz) и у Smartmouse к тому же инверсный сброс. Поэтому, если вам нужен только Phoenix, кварцевый резонатор на 6 Mhz можно не искать. Кабель от COM‑порта к интерфейсу тот же, что и для JDM.