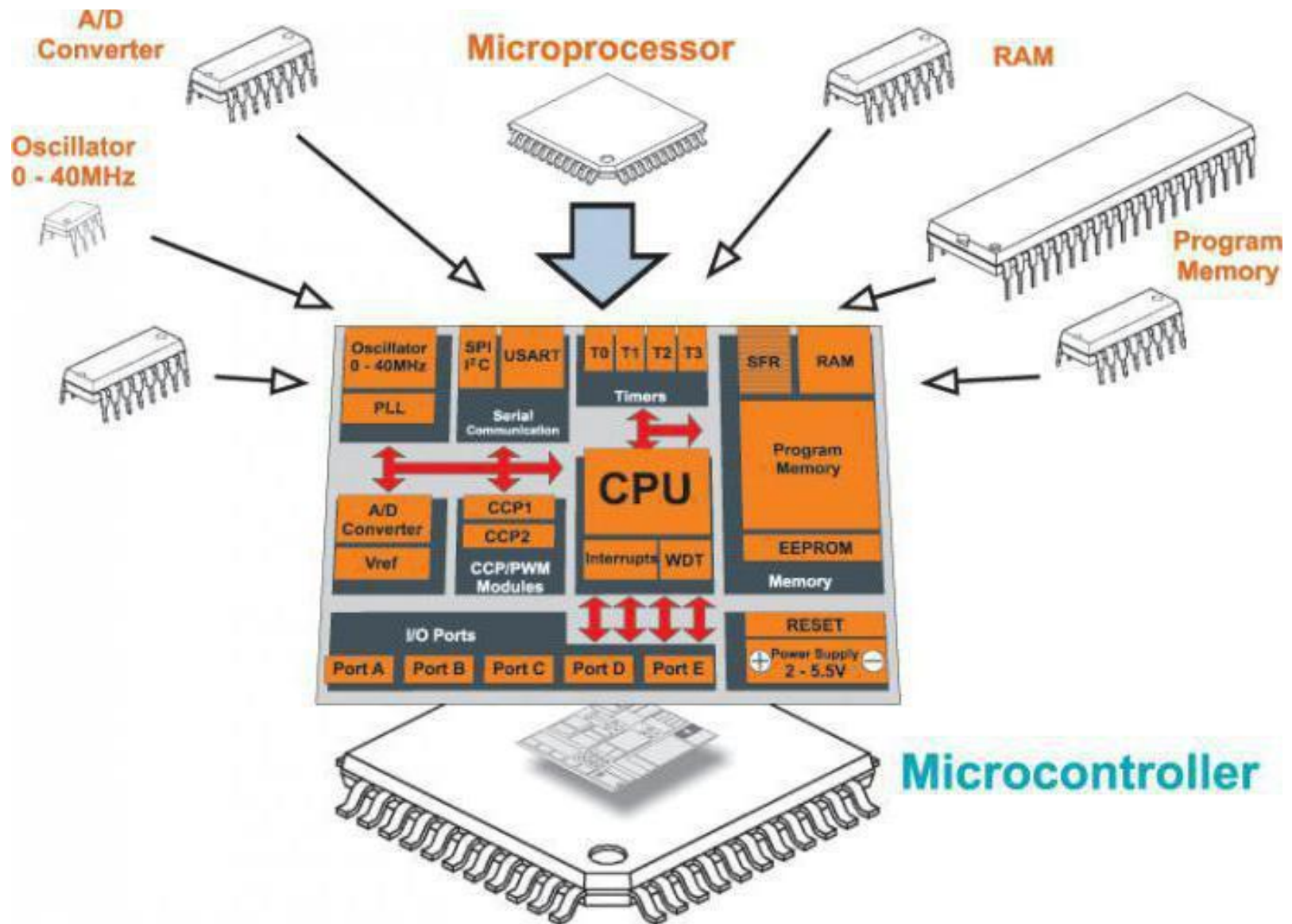


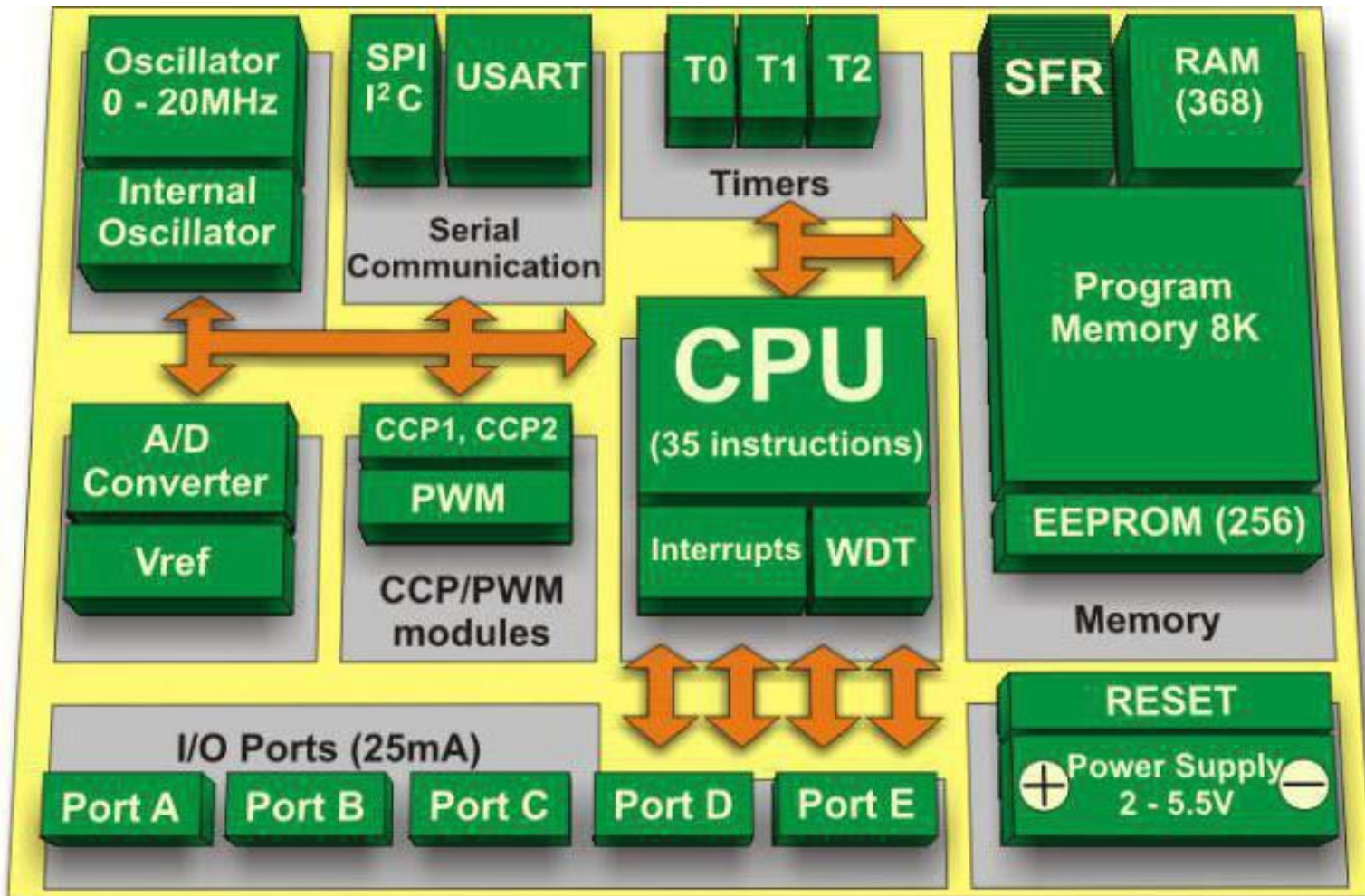


Разработка устройства основе МК

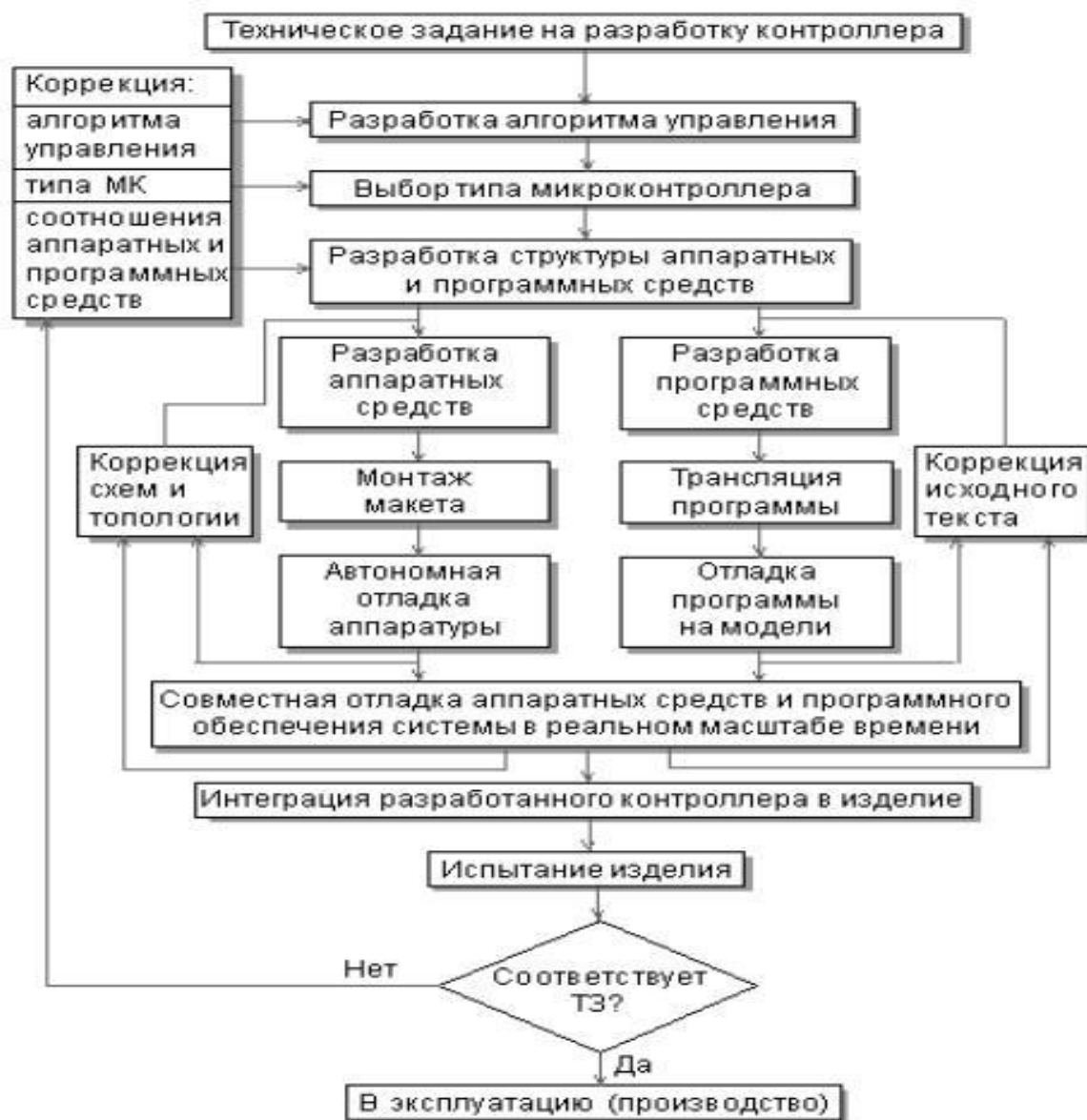
Состав микроконтроллера



Внутреннее устройство микроконтроллера PIC16



Разработка микропроцессорной системы на основе микроконтроллера. Основные этапы разработки



Языки программирования микроконтроллеров

Язык Ассемблера .

Ассемблер (от англ. assembler — сборщик) — транслятор программы из текста на языке ассемблера, в программу на машинном языке.

Язык Ассемблера (иногда его называют не совсем корректно Ассемблер) является языком самого низкого уровня. При этом он позволяет наиболее полно раскрыть все возможности микроконтроллеров и получить максимальное быстродействие и компактный код. В некоторых случаях альтернативы ассемблеру нет, но тем не менее он имеет множество недостатков. Несмотря на получаемую компактность машинного кода, программа, написанная на языке Ассемблера, громоздка и труднопонимаема. Для ее создания требуется отличное знание архитектуры и системы команд микроконтроллеров.

Язык Ассемблера отлично подходит для программирования микроконтроллеров, имеющих ограниченные ресурсы, например 8-ми битных моделей с малым объемом памяти. Для больших программ и тем более 32-разрядных контроллеров, лучше использовать другие языки, отличающиеся более высоким уровнем. Это позволит создавать более сложные и при этом понятные программы.

C/C++

Язык программирования C/C++, относится к языкам более высокого уровня, по сравнению с Ассемблером. Программа на этом языке лучше понятна человеку. Достоинством C/C++ является огромное число программных средств и библиотек, позволяющих просто создавать необходимый код. Фактически, C/C++ сегодня стал основным языком разработки управляющих программ. Компиляторы данного языка реализованы практически для всех моделей микроконтроллеров. Стандартный язык дает возможность переноса программ с одной платформы на другую. Теоретически, используя разные компиляторы, можно преобразовать любую программу в команды микроконтроллера нужного типа. На практике дополнительно требуется учитывать архитектуру микроконтроллера каждого типа. Язык C/C++ имеет достаточно сложную для изучения структуру.

Получаемый программный код конкретной задачи, имеет больший объем, чем код той же задачи, реализованной на Ассемблере. Тем не менее язык C/C++ следует признать единственным правильным выбором для профессионального программирования микроконтроллеров.

Язык C++

- **Язык C++** (произносится как «*Си плас плас*») был разработан Бьёрном Страуструпом в подразделении Bell Labs компании AT&T в качестве дополнения к языку Си. C++ добавил множество новых возможностей в язык Си. Его популярность была вызвана объектно-ориентированностью языка. Сейчас C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. С его помощью создают операционные системы, разнообразные прикладные программы, драйверы устройств, игры и пр. В настоящее время этот язык является основным языком программирования микроконтроллеров.
- Среди самого популярного программного обеспечения, написанного на C++ (или с его использованием), находятся СУБД MySQL, интернет-браузер Mozilla Firefox, большая часть программного обеспечения от Microsoft:

Выбор МК

При выборе типа МК учитываются следующие основные характеристики:
разрядность;

- быстродействие;
- набор команд и способов адресации;
- требования к источнику питания и потребляемая мощность в различных режимах; объем ПЗУ программ и ОЗУ данных;
- возможности расширения памяти программ и данных;
- наличие и возможности периферийных устройств, включая средства поддержки работы в реальном времени (таймеры, процессоры событий и т.п.);
- возможность перепрограммирования в составе устройства; наличие и надежность средств защиты внутренней информации;
- возможность поставки в различных вариантах конструктивного исполнения; стоимость в различных вариантах исполнения;
- наличие полной документации;

Задача:

- Зажечь светодиод с помощью кнопки через МК.
(Чтение порта и запись в порт - интерфейса ввода/вывода общего назначения МК (general-purpose input/output, GPIO))

Этапы решения

- Выбираем МК, читаем даташит (PDF – описание).
- Составляем схему электрическую принципиальную
- Собираем устройство согласно схеме
- Составляем блок схему алгоритм программы
- Пишем код по блок- схеме
- Програмируем контроллер (загружаем программу в МК)

Краткая характеристика выбранного

микроконтроллера

Выбираем микроконтроллер PIC16F877-04I/P, Микроконтроллер PIC 8192 x 14 - ППЗУ/368-ОЗУ 8-АЦП 33порт(-ов) ввода-вывода 3- таймера + сторожевой таймер шины USART/MSSP 2x10-бит-ШИМ ICSP. Цена 279,0 рублей

Корпус (размер)	40-DIP (0.600", 15.24mm)
Рабочая температура	-40°C ~ 85°C
Тип осцилятора	External
Преобразователи данных	A/D 8x10b
Напряжение источника (Vcc/Vdd)	4 V ~ 5.5 V
Размер памяти	368 x 8
EEPROM Size	256 x 8
Тип программируемой памяти	FLASH
Размер программируемой памяти	14KB (8K x 14)
Число вводов/выводов	33
Периферия	Brown-out Detect/Reset, POR, PWM, WDT
Подключения	I ² C, SPI, UART/USART
Скорость	20MHz
Размер ядра	8-Bit
Процессор	PIC
Серия	PIC® 16F

Микроконтроллер PIC16f877

PDIP

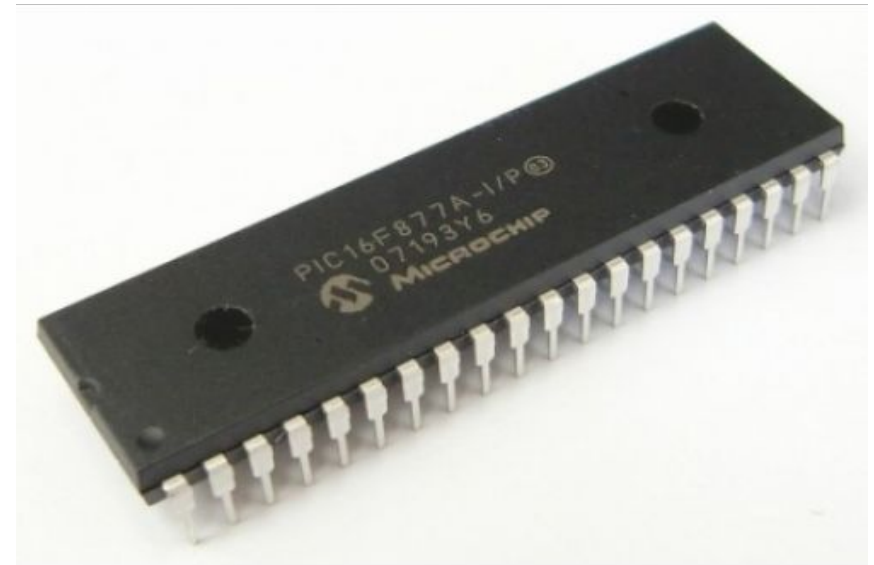
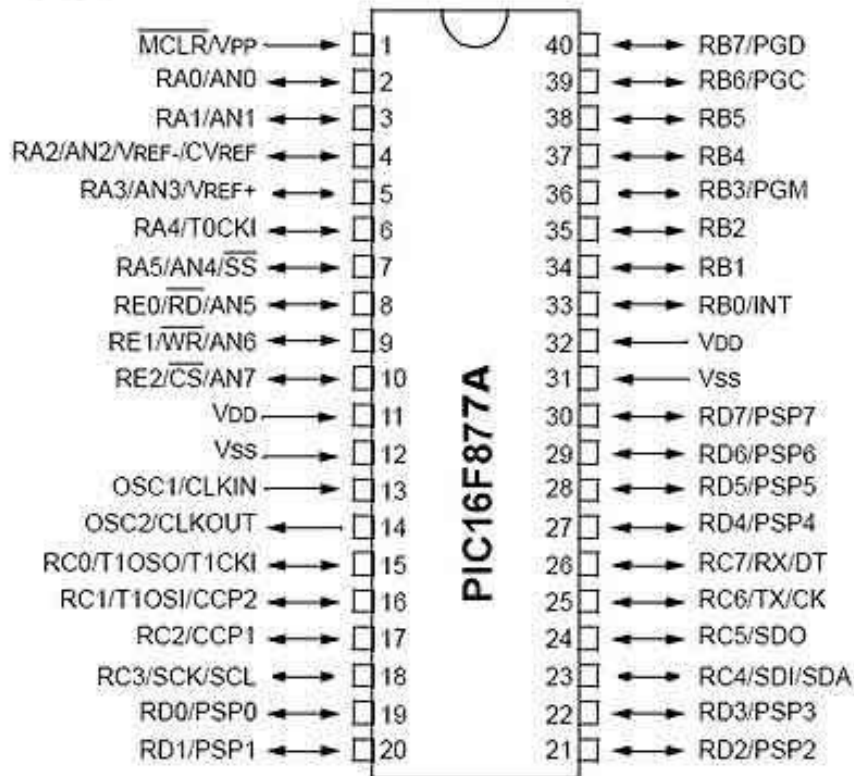
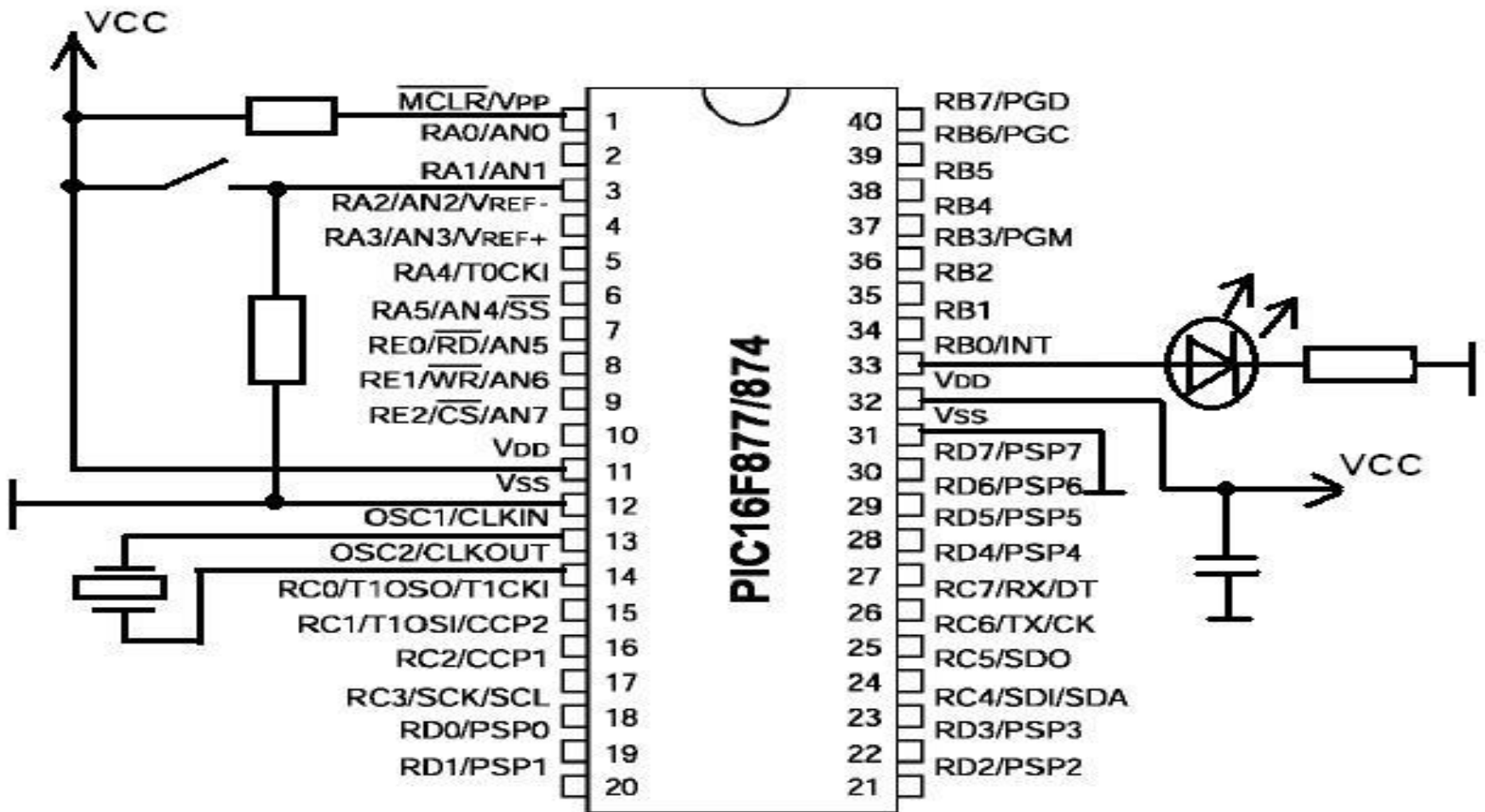
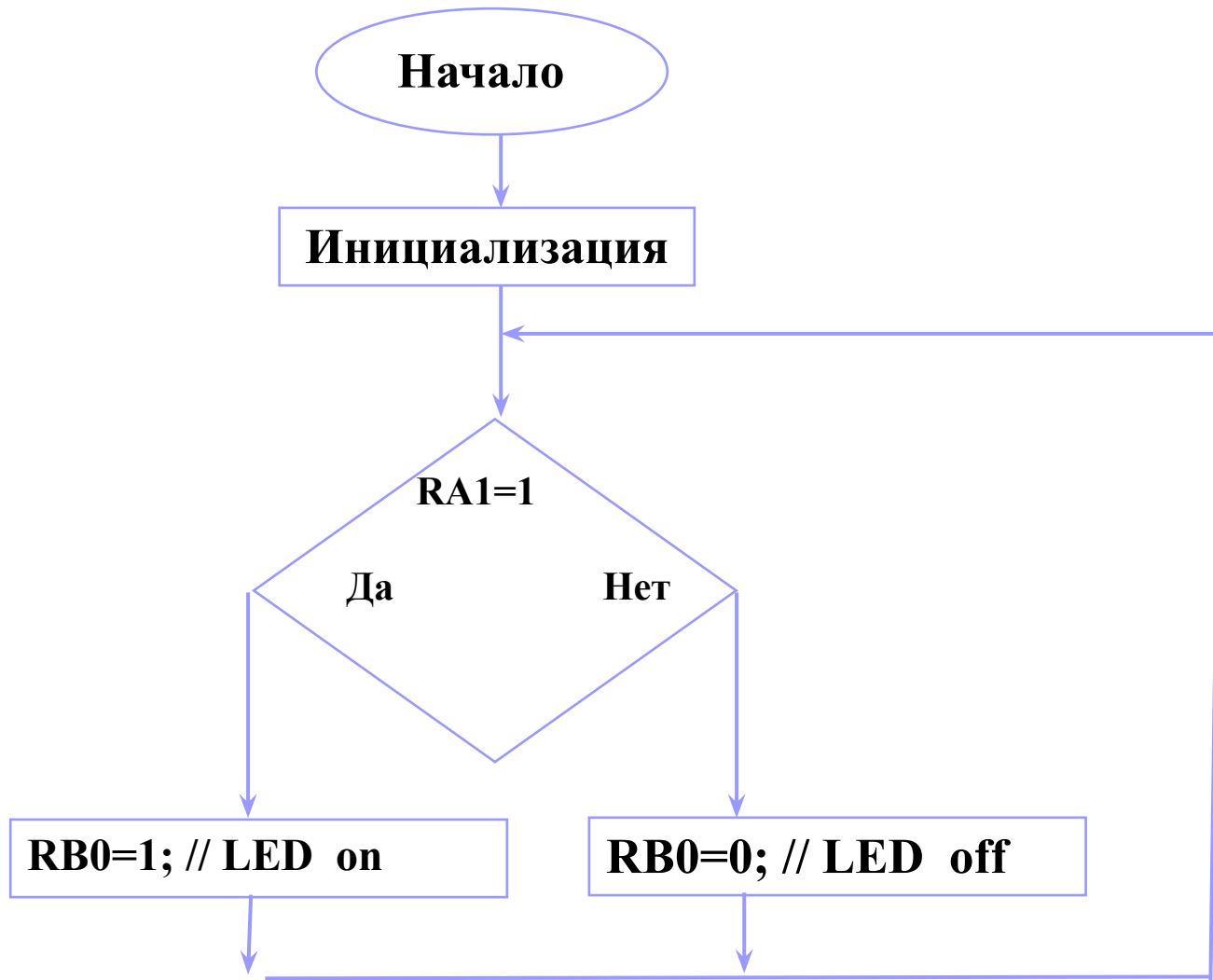


Схема устройства



Блок-схема алгоритма



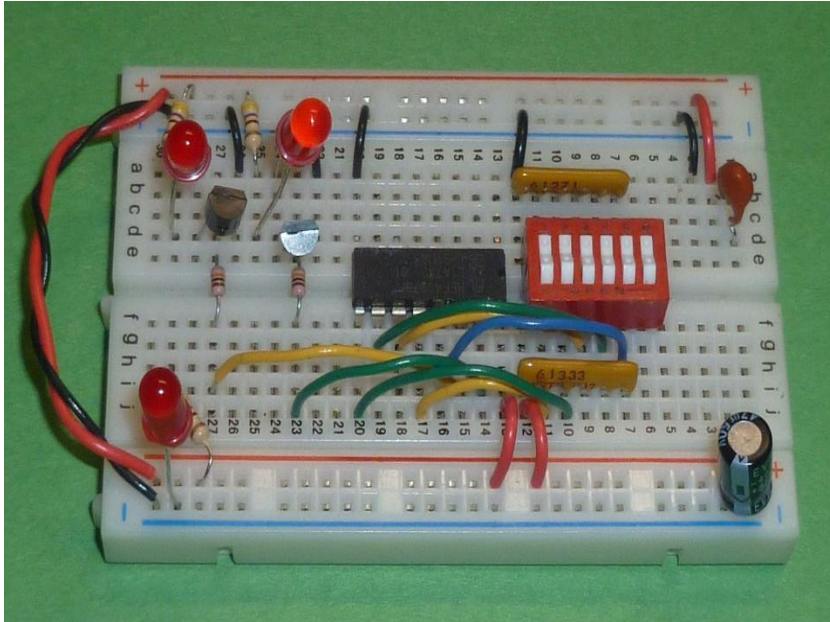
Кодирование алгоритма

```
#include <pic.h> // подключаем хидеры (библиотеки и файлы)
__CONFIG(0x03F72); // конфигурируем МК
void main(void) //начало загрузки
{
    TRISA<1>=1; // установка бита порта на вход
    // TRISA=0b111111; //выход-0 вход-1
    TRISB=0; //установка бита порта на выход
    PORTB=0; //исходная установка данных в регистрах порта
    RA1 = 0; //исходная установка данных в регистрах порта

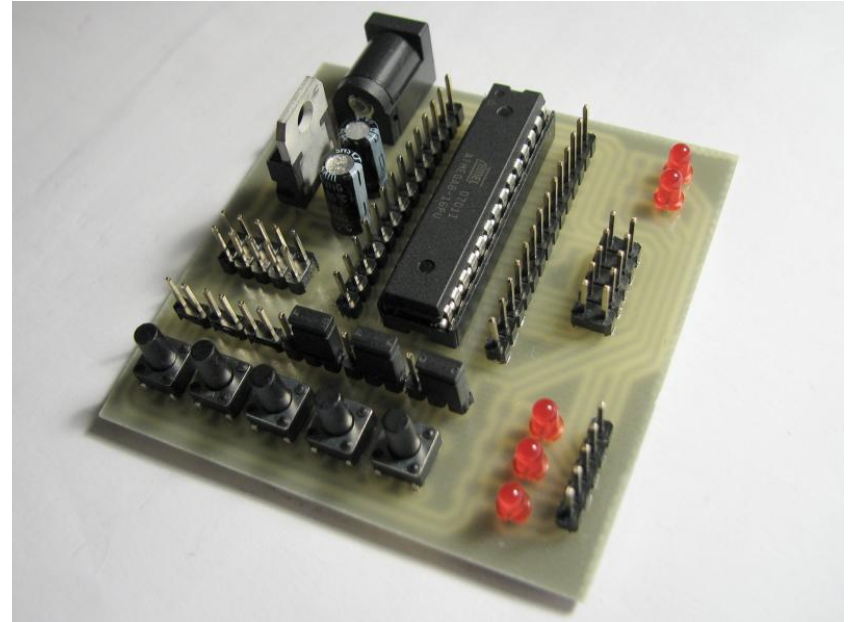
    while(1) // бесконечный цикл
    {

        if (RA1==1) //условие
            RB0=1; //да
        else RB0=0; //нет
    }
}
```

Сборка устройства



Макетная плата



Печатная плата

Загрузка программы в МК

Существует три основных варианта загрузки файла с кодом прошивки:

- 1) Программирование микросхемы через программатор с зажимной панелькой. МК нужно для прошивки ставить в панельку программатора.
- 2) Внутрисхемное программирование. Это программирование микроконтроллеров, не извлекая их с рабочей платы устройства.
- 3) Программирование через Bootloader и стандартный интерфейс. BootLoader это небольшая программа которая размещена в специальной области памяти микроконтроллера и взаимодействует с каким-либо интерфейсом. Обычно это UART, USART, но бывает и SPI и USB. При этом программы загружаются непосредственно с ПК, на котором они разрабатывались.

