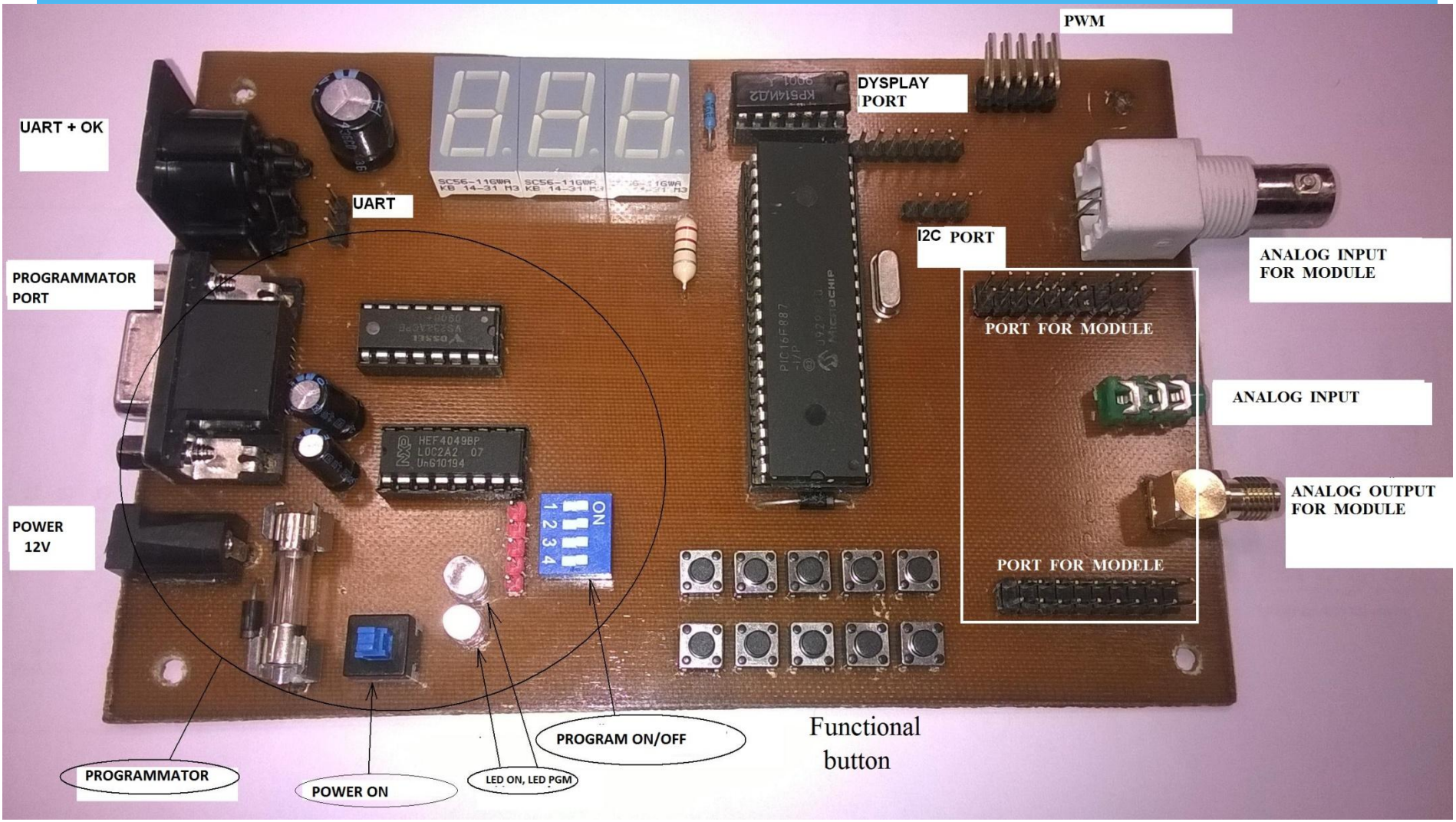
MICROCONTROLLERS BOARD
MISIS BOARD 877

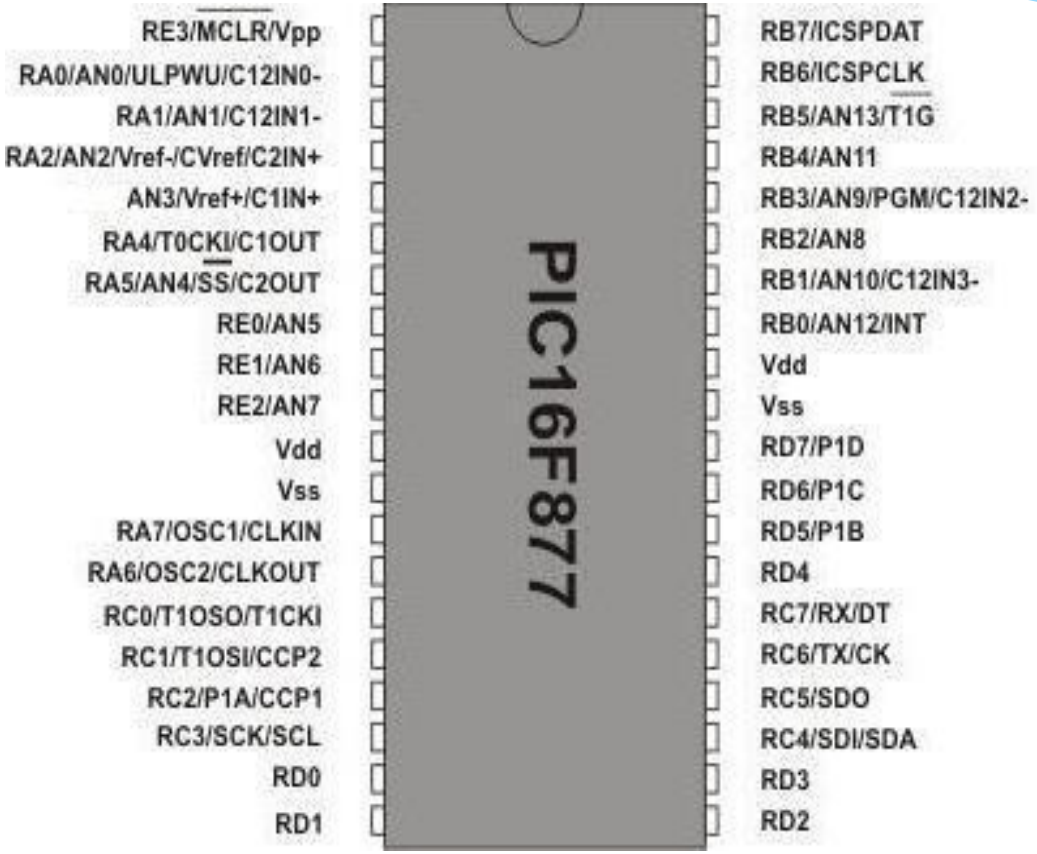
Lecture 3

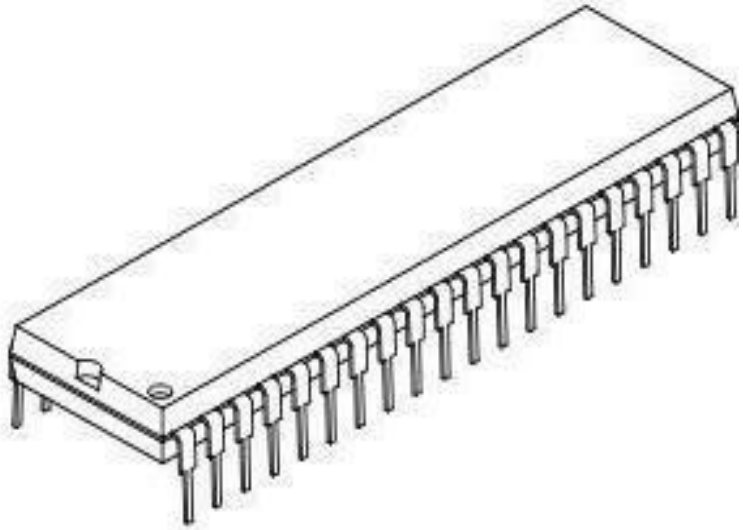
MISiS BOARD 877



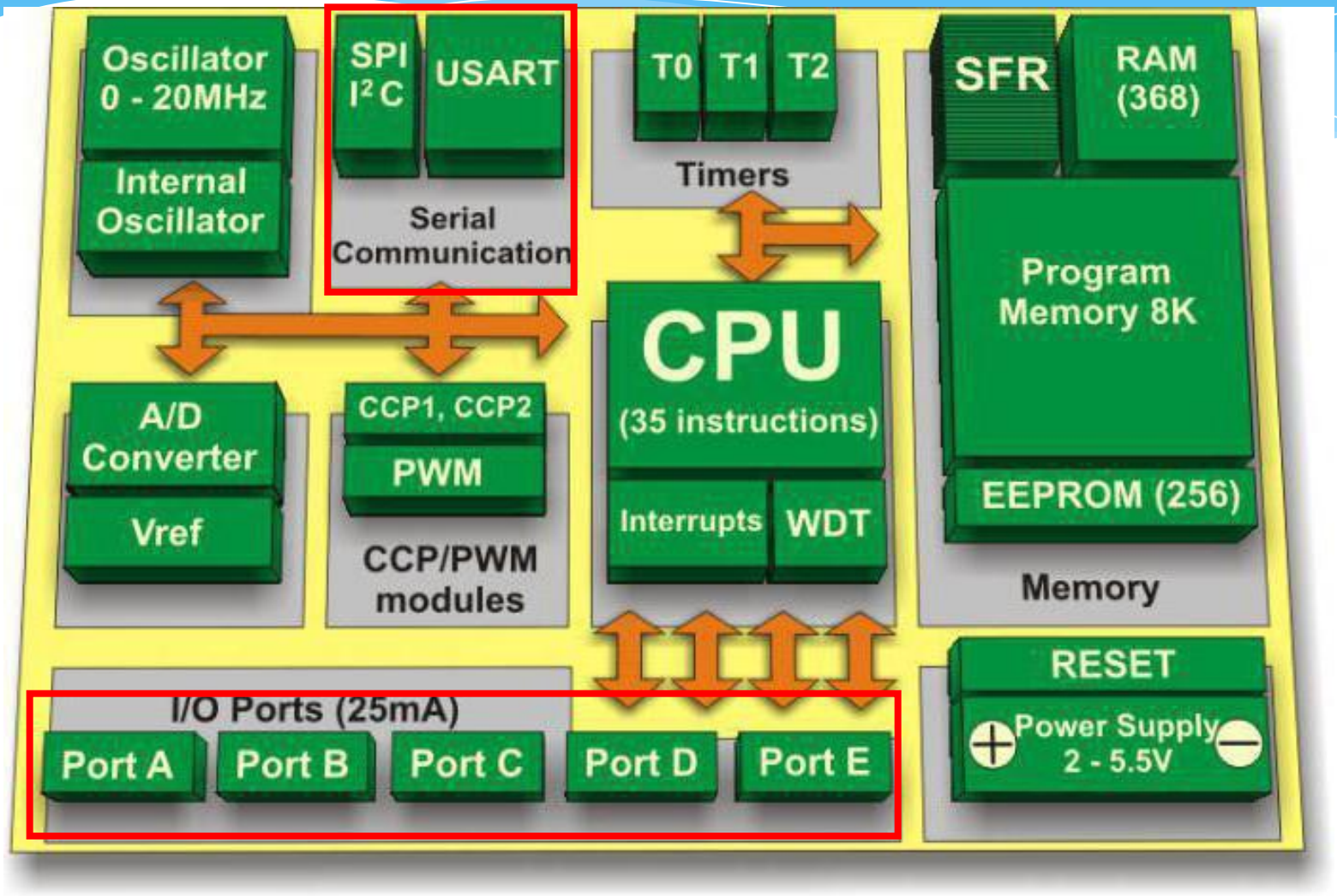
PCB

Pins interfaces

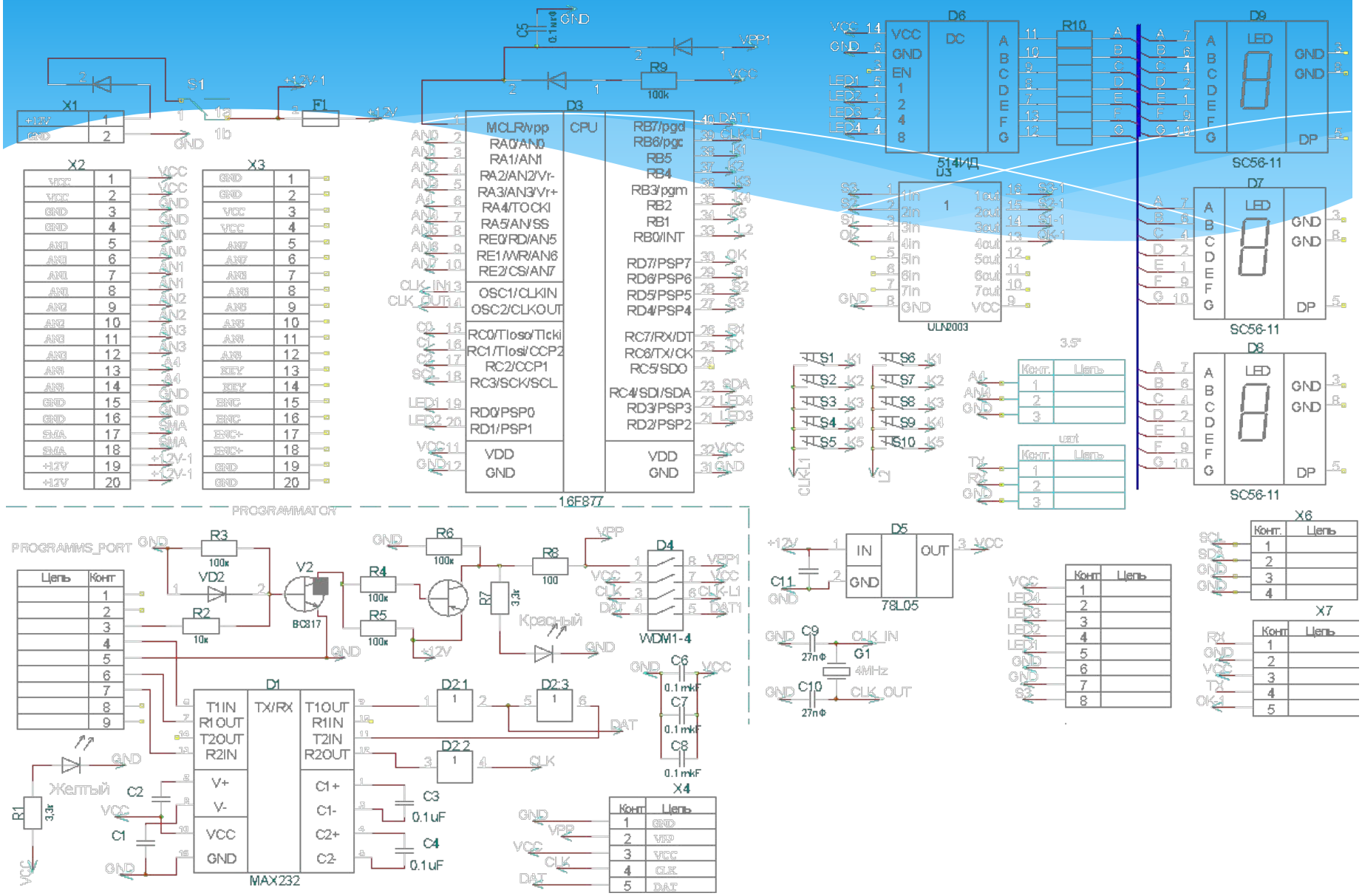




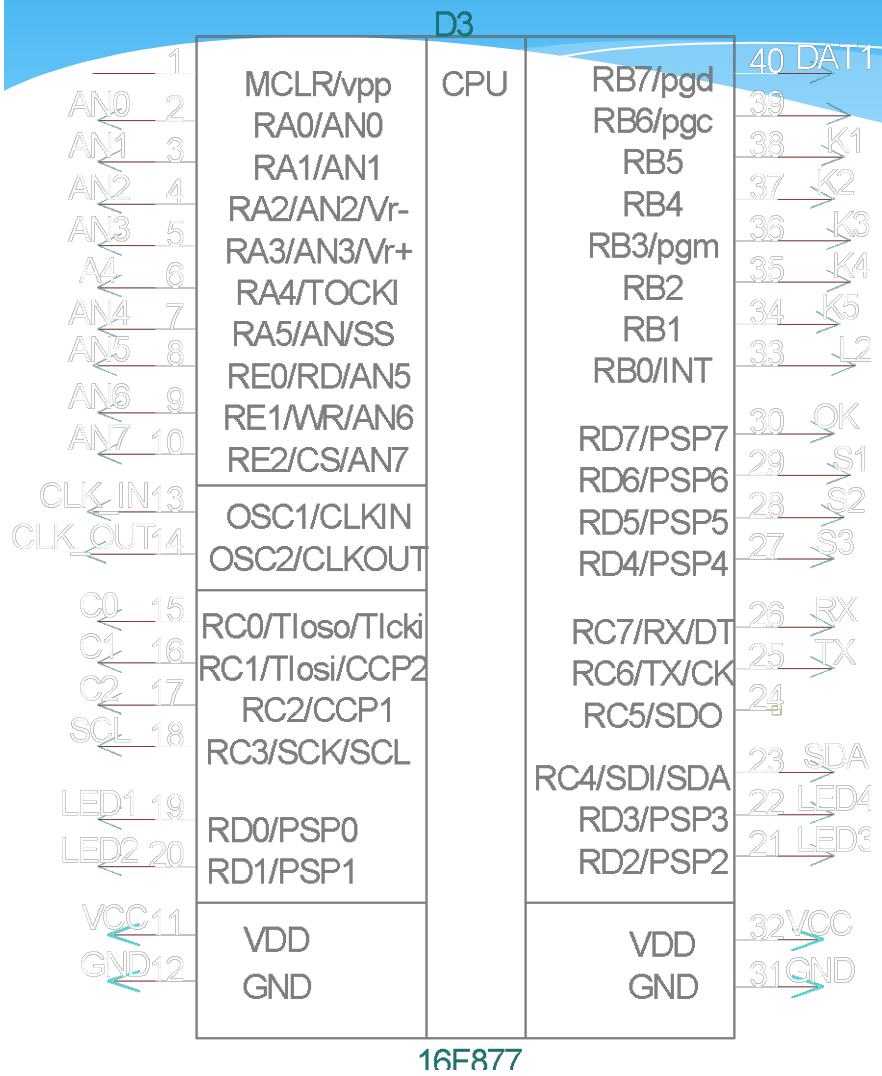
INTERFACES ON BLOCK DIAGRAM PIC16F877

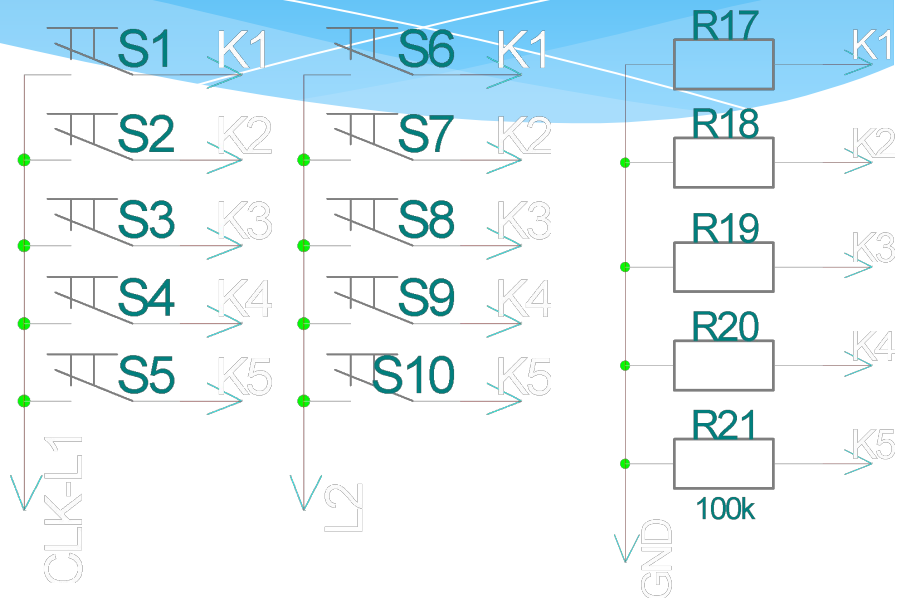


Schematic diagram



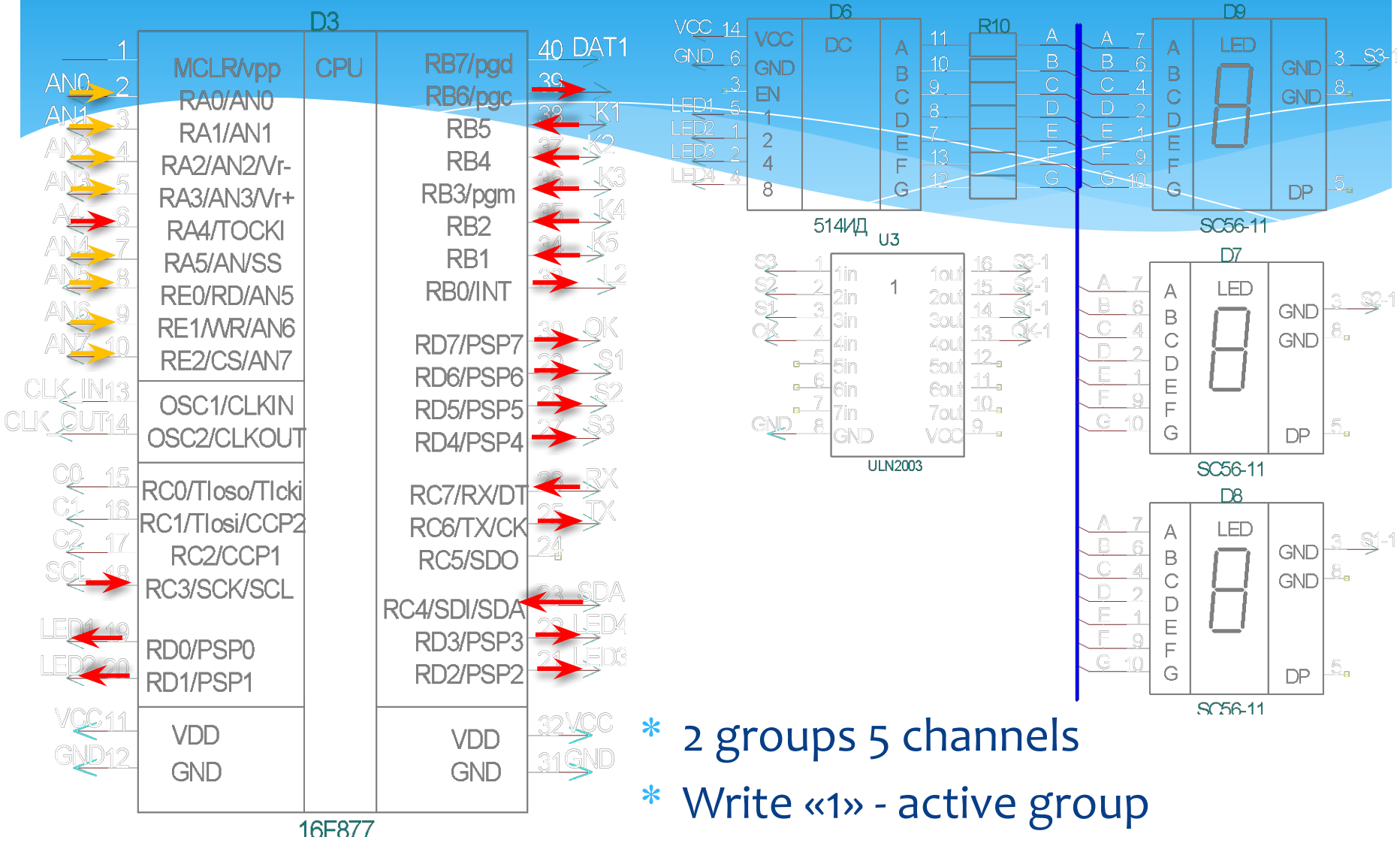
Connections button





* 2 groups 5 channels
* Write «1» - active group
* Reed «1» - touch button

Connections LED



* 2 groups 5 channels
* Write «1» - active group
* Reed «1» - touch button

Connectors

X4

Конт	Цепь
1	GND
2	VPP
3	VCC
4	CLK
5	DAT

X1

+12V	1
GND	2

X6

Конт.	Цепь
1	SCL
2	SDA
3	GND
4	GND

3.5"

Конт.	Цепь
1	A4
2	AN4
3	GND

uart

Конт.	Цепь
1	TX
2	RX
3	GND

Конт	Цепь
1	VCC
2	LED4
3	LED3
4	LED2
5	LED1
6	GND
7	GND
8	S3

X7

Конт	Цепь
1	RX
2	GND
3	VCC
4	TX
5	OK-1

From PDF (Write data in EEPROM)

1. Если шаг 10 не был выполнен, то необходимо проверить, что не происходит запись (WR=0).
2. Записать адрес в регистр EEADR. Проверьте, что записанный адрес корректен для данного типа микроконтроллера.
3. Записать 8-разрядное значение в регистр EEDATA.
4. Сбросить в '0' бит EEPGD для обращения к EEPROM памяти данных.
5. Установить бит WREN в '1', разрешив запись в EEPROM память.
6. Запретить прерывания, если они разрешены.
7. Выполнить обязательную последовательность из пяти команд:
   - Запись значения 55h в регистр EECON2 (две команды, сначала в W затем в EECON2);
   - Запись значения AAh в регистр EECON2 (две команды, сначала в W затем в EECON2);
   - Установить бит WR в '1'.
8. Разрешить прерывания (если необходимо).
9. Сбросить бит WREN в '0'.
10. После завершения цикла записи сбрасывается в '0' бит WR, устанавливается в '1' флаг прерывания EEIF (сбрасывается программно). Если шаг 1 не выполняется, то необходимо проверить состояние битов EEIF, WR перед началом записи.

1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
3. Write the 8-bit data value to be programmed in the EEDATA register.
4. Clear the EEPGD bit to point to EEPROM data memory.
5. Set the WREN bit to enable program operations.
6. Disable interrupts (if enabled).
7. Execute the special five instruction sequence: • Write 55h to EECON2 in two steps (first to W, then to EECON2) • Write AAh to EECON2 in two steps (first to W, then to EECON2) • Set the WR bit
8. Enable interrupts (if using interrupts).
9. Clear the WREN bit to disable program operations.
10. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set. (EEIF must be cleared by firmware.) If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to clear, to indicate the end of the program cycle.

Write data in EEPROM

```
void ZAP_N(void)
{
  while(WR == 1)  //wait, when ends past record
            ;          //empty string, waiting until the condition correctly,
EEADR = 204;      // choice cell EEPROM
 EEDATA = N;  //write data to register
EEPGD = 0;   // choice to memories data or programs выбор памяти данных
  WREN = 1;       // allow record
  GIE = 0;        // close all interruptions
  EECON2 = 0x55;    //small magic (from datasheet)
  EECON2 = 0xAA;
  WR = 1;         //command record
  GIE = 1;        //open all interruptions
  WREN = 0;       //prohibition record
}
```

From PDF (Read data from EEPROM)

1. Записать адрес в регистр EEADR. Проверьте, что записанный адрес корректен для данного типа микроконтроллера.
2. Сбросить в '0' бит EEPGD для обращения к EEPROM памяти данных.
3. Инициализировать операцию чтения установкой бита RD в '1'.
4. Прочитать данные из регистра EEDATA.

1. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
2. Clear the EEPGD bit to point to EEPROM data memory.
3. Set the RD bit to start the read operation. 4. Read the data from the EEDATA register.

Read data from EEPROM

```
void READ_N(void)
{
  EEADR = 204;       //адрес ячейки EEPROM
  EEPGD = 0;      //обращение к памяти данных
  RD = 1;         //запуск чтения
  N = EEDATA;  //читаем данные
}
```

personal task

№	Problem	Description input and output
1	Counter click	LED + 5 Button (+1, -1, +10, -10, reset)
2	Watch	LED + 3 Button (hour+1, minute+1, reset)
3	Timer on 500 sec (counting in reverse of time)	LED + 3 Button (reset, start, stop)
4	LED - ROM	LED + EEPROM + 3 Button (+1, -1, reset)
5	UART-monitor	Data UART to LED + 1 Button (reset)
	EXAMPLE Random digit 1-6 and send	1 Button , UART. Touch button for count digits 1-6. Don't touch one send to UART.

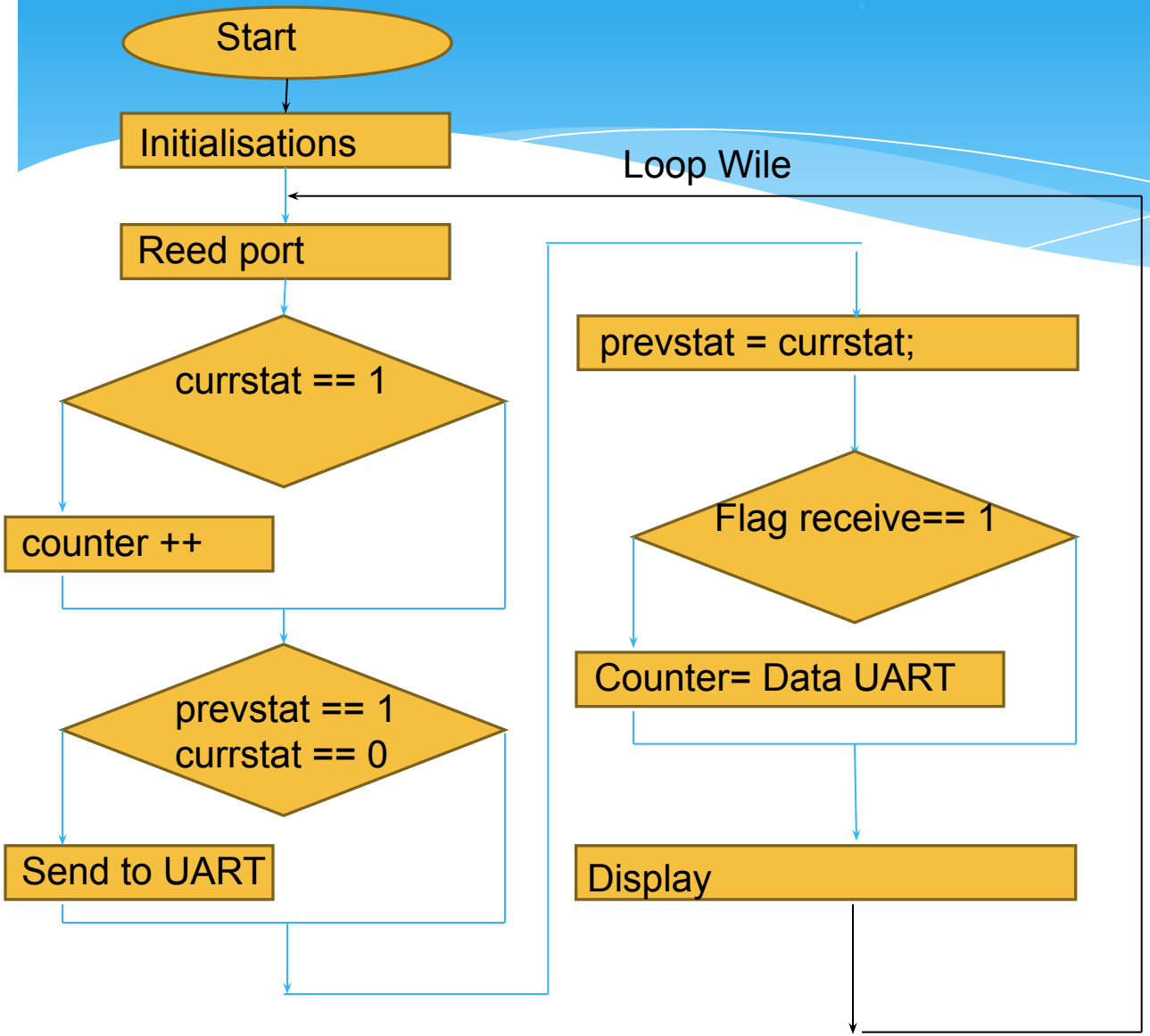
Random digit 1-6 and send UART.

1 Button , UART. Touch button for count digits 1-6. Don’t touch one send to UART.

- читаем PDF.
- Читаем схему электрическую принципиальную
- Составляем блок схему программы
- Пишем код по блок схеме
- Программируем

- **read a PDF.**
- **Read the schematic circuit diagram**
- **Draw up a block diagram of the program**
- **Write the code on the block diagram**
- **Programmable**

Block Diagram



CODE C

```
#include <pic.h>
__CONFIG(0x03F72);
char curstat; //текущее состояние
char oldstat; //старое состояние
char counter=1;

void Delay(int count)
{
        int i;
        for(i = 0; i < count; i++)
        {
                i++;
                i--;
            }
}

void SendUart(unsigned char value)
{
        while(TXIF == 0)
                ;
        TXREG = value;
}

void Display(unsigned char value)
{
PORTD= 0b01000000|counter;
}
```

CODE C

```
void main(void)
{
   TRISA=0b11110001;        //выход-0  вход-1     output-0,  input-1
   TRISB=0b01111110;       //выход-0  вход-1    output-0,  input-1
   TRISC=0b10111111;        //uart i2c и входы    output-0,  input-1  (UART, I2C pin in port)
   TRISD=0;                 //выход-0  вход-1
   TRISE=0b00000011;        //выход-0  вход-1
   PORTA=0;
   PORTB=0b01000001; // activated button
   PORTC=0;
   PORTD=0;

   while(1==1)     ////////////ОСНОВНОЙ ЦИКЛ ////////////////
   {
     curstat=PORTB&0b00111110;

     if(curstat==2)
        counter=counter+1;

     if(curstat<oldstat)
        SendUart(counter);

     oldstat=curstat;

     if(RCIF)
       counter = RCREG;

     Display();
    }
}
```

```
if(counter==7)
Counter=1;
```

НАЙДИТЕ ОШИБКУ

SEARCHING ERROR