EXPERIMENT - 2

Aim: Study and Practical implementation of Cross-Wired and Straight Through network cables.

Objective: To know the procedure of making straight and cross over network cables which are used for wired network communication.

Conceptual Background:

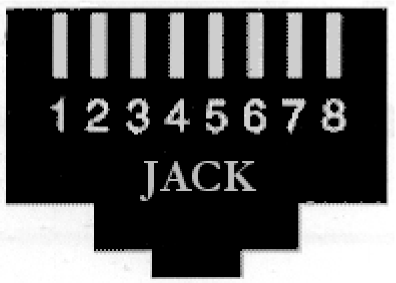
10/100BaseT Ethernet wiring is based upon 8-pin modular jacks. Two wiring guidelines are offered in Electronic Industry Association (EIA)/Telecommunications Industry Association (TIA) standards:

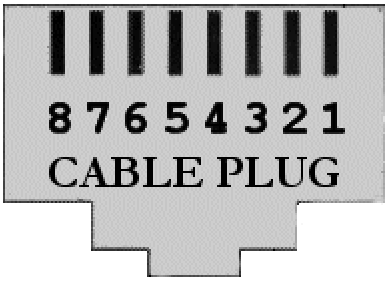
- EIA/TIA 568A
- EIA/TIA 568B

A standard Ethernet NIC card (DTE) has the following RJ-45 modular pinout:

Pin	Function
1	Tx+
2	Tx-
3	Rc+
6	Rc-

The RJ-45 modular jack has 8 conductors/pins. Viewed from the front:





What is 568?

In the world of structured cabling systems the cryptic number 568 refers to the order in which the individual wires inside a CAT 5 cable are terminated. The termination could come at either the user's end socket, the patch panel or termination frame or even the individual leads that connect a computer to the wall socket. There are currently two different specifications with respect to the order these cables should be terminated contained in the international standards document (ISO/IEC 11801:1995) as there is no indication as to which of these standards is preferred.

TIA-568A

In TIA-568A wiring schemes, the White/Orange and White/Green pairs are transposed. An Ethernet "Crossover" cable can be made by using TIA-568B at one end of the cable and wiring to TIA-568A at the other end of the cable. These "Crossover" cables can be used to interconnect NIC cards directly, without the need for a hub. "Crossover" cables are usually "**RED**" in color:

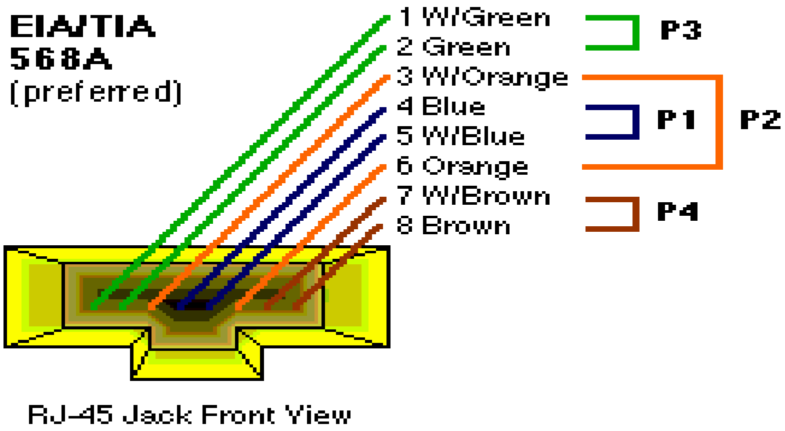
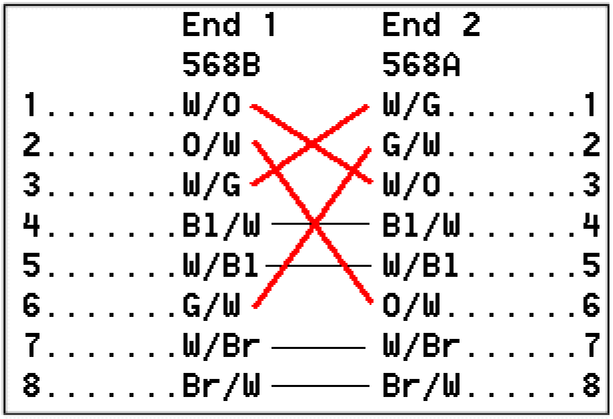


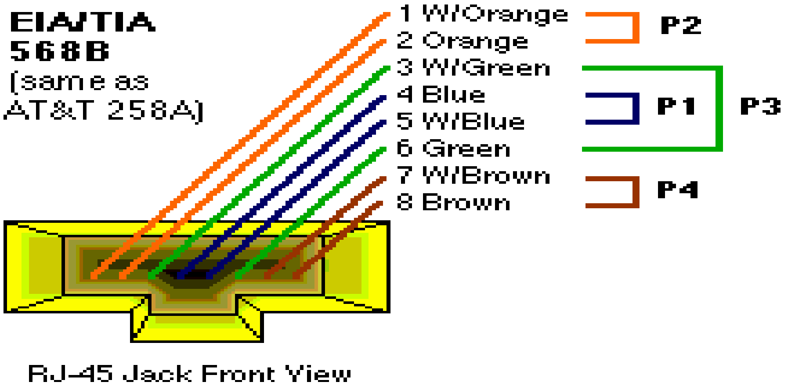
FIGURE 1.0 SHOWS HOW THE TIA/EIA 568A STANDARD IS TO BE TERMINATED. NOTE THE POSITION OF THE GREEN/WHITE GREEN AND THE ORANGE/WHITE ORANGE PAIRS.



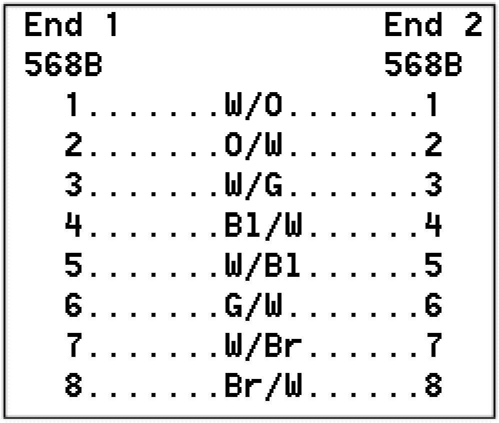
	End 1 568B	End 2 568A	
1.......	W/O	W/G.......	1
2.......	O/W	G/W.......	2
3.......	W/G	W/O.......	3
4.......	Bl/W	Bl/W......	4
5.......	W/Bl	W/Bl......	5
6.......	G/W	O/W.......	6
7.......	W/Br	W/Br......	7
8.......	Br/W	Br/W......	8

TIA-568B

TIA-568B is **the most common wiring scheme** used in 10/100BaseT configurations. The standard Ethernet "Patch" cable that connects a hub port to a NIC card uses TIA-568 wiring at both ends of a "straight-through" cable:



IN FIGURE 1.1 YOU CAN SEE THE TIA/EIA 568B STANDARD. ONCE AGAIN NOTE THE POSITION OF THE GREEN/WHITE GREEN AND ORANGE/WHITE ORANGE PAIRS.



End 1		End 2
568B		568B
1.......	W/O	1
2.......	O/W	2
3.......	W/G	3
4.......	Bl/W	4
5.......	W/Bl	5
6.......	G/W	6
7.......	W/Br	7
8.......	Br/W	8

COMPARING 568A AND 568B

By looking at the first two specifications we see that the only difference is that the green and orange pairs are terminated to different pins, there is no difference as to what signal is used on what pin, only what color wire is terminated onto it. So technically the standards are the same, they operate in the same manner and neither one is technically superior to another when used in Ethernet applications.

It is when an Ethernet system and a phone system are combined that the difference really becomes apparent.

Procedure:

PATCH CABLE ASSEMBLY INSTRUCTIONS

1. Skin the cable jacket back about an inch, possibly more.
2. Un-twist each pair, and straighten each wire between the fingers.
3. There are two common wiring standards. Before this step be sure which type you need.

If you need T568B then place the wires in this order: White/Orange, Orange, White/Green, Blue, White/Blue, Green, White/Brown, Brown

However if you need T568A then use this order; White/Green, Green. Whit/Orange, Blue, White/Blue, Orange, White/Brown, Brown. Bring all of the wires together, until they touch.

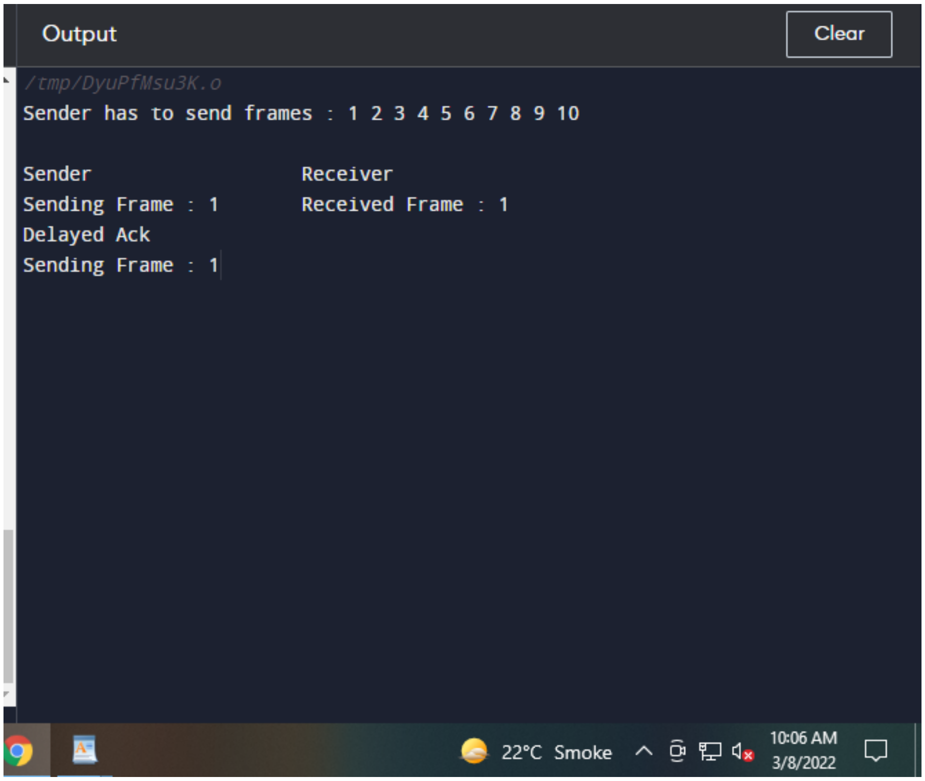
4. Recheck the wiring sequence.

5. Hold the grouped (and sorted) wires together tightly, between the thumb, and the forefinger.
6. Cut all of the wires at a perfect 90 degree angle from the cable at 1/2" from the end of the cable jacket. If the wires are not cut straight, they may not all make contact.
7. Insert the wires into the connector (pins facing up).
8. Place the connector into a crimp tool, and squeeze hard so that the handle reaches it's full swing.
9. Repeat the process on the other end. For a straight through cable, use the same wiring.
10. Use a cable tester to test for proper continuity.

Experiment : 3

```
#include<iostream>
#include <time.h>
#include <cstdlib>
#include<ctime>
#include <unistd.h>
using namespace std;
class timer {
    private:
     unsigned long begTime;
    public:
     void start() {
      begTime = clock();
     }
  unsigned long elapsedTime() {
      return ((unsigned long) clock() - begTime) / CLOCKS_PER_SEC;
    }
   bool isTimeout(unsigned long seconds) {
      return seconds >= elapsedTime();
     }
};
int main()
{
 int frames[] = {1,2,3,4,5,6,7,8,9,10};
 unsigned long seconds = 5;
 srand(time(NULL));
 timer t;
 cout<<"Sender has to send frames : ";
 for(int i=0;i<10;i++)
     cout<<frames[i]<<" ";
 cout<<endl;
 int count = 0;
 bool delay = false;
 cout<<endl<<"Sender\t\t\t\t\tReceiver"<<endl;
```

```
do
{
    bool timeout = false;
    cout<<"Sending Frame : "<<frames[count];
    cout.flush();
    cout<<"\t\t";
    t.start();
    if(rand()%2)
    {
        int to = 24600 + rand()%(64000 - 24600)  + 1;
        for(int i=0;i<64000;i++)
            for(int j=0;j<to;j++) {}
    }
    if(t.elapsedTime() <= seconds)
    {
        cout<<"Received Frame : "<<frames[count]<<" ";
        if(delay)
        {
            cout<<"Duplicate";
            delay = false;
        }
        cout<<endl;
        count++;
    }
    else
    {
        cout<<"---"<<endl;
        cout<<"Timeout"<<endl;
        timeout = true;
    }
    t.start();
    if(rand()%2 || !timeout)
    {
        int to = 24600 + rand()%(64000 - 24600)  + 1;
        for(int i=0;i<64000;i++)
            for(int j=0;j<to;j++) {}
        if(t.elapsedTime() > seconds )
        {
          cout<<"Delayed Ack"<<endl;
          count--;
          delay = true;
        }
        else if(!timeout)
          cout<<"Acknowledgement : "<<frames[count]-1<<endl;
    }
 }while(count!=10);
 return 0;
}
```



```
/tmp/DyuPfMsu3K.o
Sender has to send frames : 1 2 3 4 5 6 7 8 9 10

Sender                  Receiver
Sending Frame : 1       Received Frame : 1
Delayed Ack
Sending Frame : 1
```