

---

## TCPDAQ data structure

---

### TCPDAQ Data Structure

```
typedef struct _AlarmInfo //Alarm Event data structure
{
    u_char      szIP[4];           //The IP address which cause the alarm change
    u_short     szDateTime[6];    //E.x 2001/09/23 10:12:34:567 (Year/Month/Day Hour:Minute:Second:mSecond)
    u_short     byChannel;        //The Channel of which cause the alarm change
    u_short     byAlarmType;      //0x00:AIO Low Alarm
                                        //0x01:AIO High Alarm
                                        //0x20:DIO Alarm
                                        //0xF0:Connection Alarm
    u_short     byAlarmStatus;    //0:Alarm ON to OFF, 1:Alarm OFF to ON
    u_short     wValue;          //Alarm value.For DIO, this value could be "0" or "1" means that "ON" or "OFF"
                                        //          For high or low alarm, this is the AIO value.
                                        //          For connection lost, this value is '0'.
} _AlarmInfo;
```

```
typedef struct _StreamData //Stream Event data structure
{
    u_char      szIP[4];           //The IP address which send the stream datae
    u_short     szDateTime[6];    //E.x [2001]/[09]/[23] [10]:[12]:[34] (Year/Month/Day Hour:Minute:Second)
    u_short     DIN;              //Digital input data (DI#0~DI#15)
    u_short     DOUT;             //Digital output data (DO#0~DO#15)
    u_short     wData[32];        //Digital input Counter (Each channel occupie 4 Byte)
} _StreamData;
```

```
typedef struct ModuleInfo // Used For Scan_Online_Modules(..)
{
    u_char      szIP[4];           //IP address
    u_char      szGate[4];        //Gateway
    u_char      szMask[4];        //Submask
    u_char      szDHCP;           //DHCP status 01=enable, 00=disable
    u_char      szID;             //Module ID number
    u_char      szMacAddr[6];     //MAC address of module
    u_short     szModuleNo;       //Module name
    u_char      szBuffer[12];     //Buffer reserved for TCPDAQ.DLL
} ModuleInfo;
```

```
typedef struct ModuleData //Used for function TCP_ReadAllDataFromModule (..)
{
    u_char      Din[16];           //Digital input data (DI#0~DI#15),avaliable for EDAM9050/51/52
    u_char      Dout[16];         //Digital output data (DO#0~DO#15),avaliable for EDAM9050/51/52/17/19
    u_char      DiLatch[16];      //Digital input latch status (DI#0~DI#15),avaliable for EDAM9050/51/52
    long        DiCounter[16];    //Digital input counter value (DI#0~DI#15),avaliable for EDAM9050/51/52
    double      AiNormalValue[16]; //Analog Input value(AI#0~AI#15),avaliable for EDAM9015/17/19
    double      AiMaxValue[16];   //Analog maximum value(AI#0~AI#15),avaliable for EDAM9015/17/19
    double      AiMinValue[16];   //Analog minimum value(AI#0~AI#15),avaliable for EDAM9015/17/19
    u_char      AiHighAlarm[16];  //Analog high alarm status(AI#0~AI#15),avaliable for EDAM9015/17/19
    u_char      AiLowAlarm[16];   //Analog low alarm status(AI#0~AI#15),avaliable for EDAM9015/17/19
    u_char      AiChannelType[16]; //Analog channel Type, avaliable for EDAM9015/17/19
    u_char      AiBurnOut[16];    //Analog channel burn out status,avaliable for EDAM9019/15 only
    double      CJCTemperature ;  //Cold junction temperature,avaliable for EDAM9019 only
} ModuleData;
```