2024

TP7-LC Digital Controller

PROGRAMMER GUIDE

Ethernet Interface:

The ethernet interface is based on standard UDP protocol.

Data Frame (Packet):

All the communication with the software will done using a standard data frame:

SOF: Start of frame is always equal to 2

Len: Total frame length

CMD: Command

D: Data

LRC: LRC value for error handling (refer to LCR formula)

*Each box is represented of one byte.

By sending a frame device at the first step calculate the LRC and check the equality with received LRC to be sure that the data is not corrupted. then recognize the required action by frame command value and send specified answer. there is two type of frame, the frame for an action for example start test, in this case device send back the received frame as an answer so software can check the received frame which should be as same as sent one. The other type of frame is for reading or writing data and settings value so for reading frame, device send back specific answer according to the received command. In all condition it is better to calculate the LRC and checked with received LRC to be sure that data is not corrupted.

LRC Formula:

It is a standard method for determine the data corruption. For calculate LRC, you have to XOR bytes of the frame and the final result will be LRC. Note that BG and LRC should not use in LRC calculation.

LRC=Len(XOR)CMD(XOR)D0(XOR)Dn

For example, in test frame LRC is equal to 52:

LRC=4(XOR)48=52

Data Structures:

- **Channel: 56 Byte in total: for read channel value and system status.**
    - **Byte Status***
    - **Byte Event****
    - **Float CH[0]**
    - **Float CH[1]**
    - **Float CH[2]**
    - **Float CH[3]**

    Read: 02 04 0x22 LRC

```
* enum status
{
    Stop,
    Ramp_Up,
    Ramp_Down,
    Jog_Up,
    Jog_Down,
    PID_Mode,
    PID_Ramp,

};

** enum ev
{
    Stop,
    Run,
    PID,
    Up_Limit,
    Down_Limit,
    Load_Limit
};
```

- **System: 44 Byte in total:**
    - **byte Out_Type;**
    - **byte Inverse;**
    - **byte Monitor;**
    - **reserve one byte**
    - **uint32 ACC_Time;**
    - **uint32 Pulse_rev;**
    - **uint32 Max_Disp;**
    - **uint32 Auto_Off;**
    - **float Gear_Ratio;**
    - **float Pitch;**
    - **uint32 Max_Speed;**
    - **uint32 Jog_Speed;**
    - **float Disp_Coff;**

- **uint32 Max_Load;**

Read: 02 04 0x34 LRC

Write: 02 48 0x35 DT0…DT44 LRC

- **PID: 45 Byte in total:**
  - **float SP;**
  - **float PV;**
  - **float kP;**
  - **float kI;**
  - **float kD;**
  - **float FF;**
  - **float Err_Band;**
  - **signed int Max;**
  - **signed int Min;**
  - **unsigned int Ts;**
  - **signed int Out;**
  - **unsigned char Mode=0;**

Read: 02 04 0x3A

Write: 02 49 0x3B Dt0…Dt44 LRC

- **Sensor: 36 Byte in total for each channel: send and receive 4 channel in one 148 byte frame:**
  - **char Name[10]:** hold channel name (10 character)
  - **char SUnit[6]:** hold channel unit (6 character)
  - **unsigned char DecPoint:** Channel decimal point value
  - **unsigned char Log:** Channel log enable (0: Disable , 255: Enable)
  - **unsigned char H_Alarm:** High value alarm
  - **unsigned char L_Alarm:** Low alarm value
  - **float H_limit:** Maximum value of the channel
  - **float L_limit:** Minimum value of the channel
  - **unsigned char Filter:** Filter degree
  - **unsigned char Gain:** Gain of input amplifier (1X, 2X, 4X, 8X, 16X, 32X, 64X, 128X)
  - **unsigned char reserved_1**
  - **unsigned char reserved_2**
  - **enum calib_list CalType:** Calibration method (4byte , uint32)*

Read: 02 04 0x3C LRC

Write: 02 148 0x3D Dt0…Dt143 LRC

```
* enum calib_list
   {
         CTI_RAW,
         CTI_mVolt,
         CTI_Equation,
         CTI_Strain,
         CTI_K_Type,
         CTI_T_Type,
         CTI_RTD,
         CTI_Table,
   };
```

- **Test: 16 Byte in total:**
  - float Target[2];
  - float Speed[2];

  Read: 02 04 0x30 LRC

  Write: 02 20 0x31 LRC

- **ADC: 120 Byte in total: one for each device:**

  - **char ActiveCH[8]:** Active/De-active channel (0: De Active , 255: Active)
  - **char FilterType;**
  - **char PostFilter[8];**
  - **char Gain[8];**
  - **uint16_t FS;**
  - **char IntOC;**
  - **char IntFC;**
  - **char SysOC;**
  - **double Cofficient[8];**
  - **double K1;**
  - **double K2;**
  - **int Max;**

  Read: 02 04 0x38 LRC

  Write: 02 124 0x39 LRC

Command Table:

Command	Function	Packet Length	Answer Length	Description
0x22	Read Channel Data	4	4+18	value of all channel + status + Event
0x20	Set Channel to Zero	5	5	Answer same as sent packet
0x21	Set Pulse to Zero	4	4	Answer same as sent packet
0x3C	Read Sensor Structure	4	4+144	Sensor Structure X4
0x3D	Write Sensor Structure	4+144	4+144	Write Sensor Structure
0x38	Read ADC Structure	4	4+116	ADC Structure
0x39	Write ADC Structure	4+116	4+116	Write ADC Structure
0x3A	Read PID Structure	4	4+45	PID Structure
0x3B	Write PID Structure	4+45	4+45	Write PID Structure
0x30	Read Test Structure	4	4+16	ADC Structure
Ox31	Write Test Structure	4+16	4+16	Write ADC Structure
0x10	Motor Stop	4	4	Stop Actuator (PID or Pulse)
0x11	Jog Up	4	4	Moving Up with System.Jog Speed
0x12	Jog Down	4	4	Moving Down with System.Jog Speed
0x13	Ramp Up	4	4	Moving Up with Test.Speed[1]
0x14	Ramp Down	4	4	Moving Down with Test.Speed[1]
0x15	PID Target	4	4	Constant Load Control (SP= Target[0])
0x16	PID Ramp	4	4	Load Ramp Up (SP=Target[0] , Rate=Speed[0])
0x17	Displacement Run	4	4	Displacement control Run (SP=Target[1] , Rate=Speed[1])
0x34	Read System Structure	4	4+44	System structure
0x35	Write System Structure	4+44	4+44	Write System Structure