```
//Arquivo da TAD protocolo de comunicação entre o computador e o controlador
#ifndef PROTOCOL_H_
#define PROTOCOL_H_

#define NUM_ACT 5
#define NUM_ENC 5
//#define NUM_FLEX 2


#define CMD_POSITION 1
#define CMD_SPEED 2
#define CMD_TORQUE 3
#define CMD_GET_UP_SEQUENCE 4


#define STATUS_STOPPED 1
#define STATUS_STAND_UP 2
#define STATUS_CONTROLED 3


// COMMAND PACKAGE
typedef struct leo2_command_package{
 unsigned char header_1 ;
 unsigned char header_2 ;

 unsigned char length ;

 unsigned char command ;

 unsigned char data[2*NUM_ACT] ;
```

```
 unsigned char checksum ;
 }leo2_command_package_t ;



// STATUS PACKAGE

typedef struct leo2_status_package{
 unsigned char header_1 ;
 unsigned char header_2 ;

 unsigned char status ;

 unsigned char length ;

 unsigned char pos[2*NUM_ENC] ;

 unsigned char speed[2*NUM_ENC] ;

 //unsigned char voltage[2*NUM_FLEX] ;

 unsigned char checksum ;
 }leo2_status_package_t ;
```

```
inline void word_to_network(unsigned int word, unsigned char *data)
{

  data[0] = word&255;
  data[1] = (word>>8)&255;
}

inline unsigned int network_to_word(unsigned char *data)
{
  return data[0] + data[1]*256;
}

#endif // PROTOCOL_H_
```