```
//ComInterface.cpp
#include "ComInterface.h"
#include "Locomotion.h"
//Captures address and size of struct
//void ComInterface::begin(uint8_t * ptr, uint8_t length, NewSoftSerial *theSerial){
//address = ptr;
//size = length;
//_serial = theSerial;
//}

//Sends out struct in binary, with header, length info and checksum
/*void ComInterface::sendData(){
  uint8_t CS = size;
  _serial->print(0x06, BYTE);
  _serial->print(0x85, BYTE);
  _serial->print(size, BYTE);
  for(int i = 0; i<size; i++){
    CS^=*(address+i);
    _serial->print(*(address+i), BYTE);
  }
  _serial->print(CS);

}
*/

ComInterface::ComInterface(int thePacketType, uint8_t dataSize) : packetType(thePacketType),
size(dataSize) {}


boolean ComInterface::sizeCorrect(NewSoftSerial *_Serial)
{
    rx_len = _Serial->read();
        //Serial.println((int)size);
        Serial.print("rx_len: ");
        Serial.println((int)rx_len);
    if(rx_len != size)
    {       Serial.println("Size FAIL rx_len ==:");
                Serial.println((int)rx_len);
                Serial.println("Size ==:");
                Serial.println((int)size);
        rx_len = 0;

        return false;
    }
    return true;
}

boolean ComInterface::chkSumCorrect()
{
    if(rx_len == (rx_array_inx-1))
    {
        //seem to have got whole message
        //last uint8_t is CS
        calc_CS = rx_len;
        for (int i = 0; i<rx_len; i++)
        {
            calc_CS^=rx_array[i];
        }
        if(calc_CS == rx_array[rx_array_inx-1])
        {
            return true;
        }
                Serial.print("CS: ");
                Serial.println((int)calc_CS);
    }
        Serial.println("CheckSum FAIL");
    return false;
}

boolean ComInterface::getData(NewSoftSerial *_Serial)
{
        //Serial.println("getData");
    if(rx_len != 0)
    {
        while(_Serial->available() && rx_array_inx <= rx_len){
                        //Serial.println((int)rx_array_inx);
            rx_array[rx_array_inx++] = _Serial->read();
        }
                //Serial.println("GetData GOOD");
        return true;
    }
        Serial.println("GetData FAIL");
    return false;
}


boolean ComInterface::receiveData(NewSoftSerial *_Serial){
    if(rx_len == 0){
        if(_Serial->available() >= 3){
            while(_Serial->read() != 0x06)
            {/* Wait*/}
            switch(_Serial->read())
            {
                case LOCOMOTION_TYPE:
                    if(!sizeCorrect(_Serial))
                    {   // Whoops, data must be currupted...
                        return false;
                    }
                    if(getData(_Serial))
                    {
                        if(chkSumCorrect())
                        {// Woot we have data and its correct!!!!
                                                         //Serial.println("Setting DATA:");

                                                         //Serial.println((int)((Locomotion*)this
                                                         )->getSpeed());
```

```
                            // data stored in rx_array
                            //((Locomotion*)this)->setSpeed(10);
                            //((Locomotion*)this)->setDirectionForward(true);

                                                          rx_len = 0;
                                                  rx_array_inx = 0;
                            int sizeSpeed = sizeof(((Locomotion*)this)->addressSpeed());
                                                        //Serial.println(sizeSpeed);
                                memcpy(((Locomotion*)this)->addressSpeed(),rx_array,sizeSpeed);

                            int sizeIsForward = sizeof(*(((Locomotion*)this)->addressIsForward
                            ()));
                            memcpy(((Locomotion*)this)->addressIsForward(),rx_array+sizeSpeed,
                            sizeIsForward);

                                                       //Serial.println((int)((Locomotion*)this)-
                                                       >getSpeed());
                                                       //Serial.println((int)rx_array[0]);
                                                       //Serial.println((int)rx_array[1]);
                                                       //Serial.println((int)rx_array[3]);
                                                       //Serial.println((int)rx_array[4]);
                                                      //  Serial.println((int)rx_array[5]);
                            return true;
                        }
                    }
                case SWITCH_TYPE:
                    if(!sizeCorrect(_Serial))
                    {
                        return false;
                    }
                    default:
                    return false;
            }
        }
    }
    return false;
}
```