# SmartDRIVE – protokół transmisji szeregowej RS-485

**Dokumentacja przygotowana przez firmę Gryftec w oparciu o oryginalną dokumentację dostarczoną przez firmę Westline**

# 1. Przegląd

Kontrolery z rodziny SmartDRIVE są wyposażone w interfejs zawierający dwa porty: port szeregowy RS485 wspierający transmisję pomiędzy kilkoma równolegle połączonymi kontrolerami oraz port szeregowy *kompatybilny* z RS232 umożliwiający połączenie z pojedynczym sterownikiem, z wykorzystaniem poziomów sygnałów TTL. Dostęp do tych portów jest wzajemnie wykluczający się, ponieważ dzielą one tę samą linię komunikacyjną we wnętrzu sterownika.

Gdy żadna jednostka nie nadaje linia komunikacyjna znajduje się w stanie nieaktywnym **1**. Przejście do stanu aktywnego następuje w momencie wysłania przez nadawcę sygnału **0**, stan ten jest dominujący.

## 1.1 Zasady

Podstawowe zasady protokołu:

- bazuje na relacji master-slave, tylko master ma możliwość zainicjowania komunikacji
- na jedno połączenie fizyczne przypada jeden master i zero lub kilka slave'ów (tylko jeden dla RS232), każdy slave posiada unikalny 7-bitowy adres
- master otrzymuje całą transmisję w formie echa
- slave musi potwierdzić każde wysłane do niego polecenie w interwale 20-50ms; master musi się liczyć możliwością timeoutu i nieobecności slave'a
- parametry komunikacji wynoszą 115200 Baud z 8 bitami danych, 1 bitem stopu i brakiem bitu parzystości; dane są typu binarnego, wszystkie wartości są z zakresu od 0 do 255

## 1.2 Format wiadomości

Master transmituje ramkę polecenia zawierającą adres docelowego slave'a; wywołany slave musi odpowiedzieć za pomocą ramki odpowiedzi z określonym opóźnieniem (ustalonym przez mastera). Jeśli polecenie jest adresowane do wszystkich podłączonych slave'ów, bajt adresu musi wynosić 0 i nie występuje żadna ramka odpowiedzi.

Ramki muszą być transmitowane blokami. Wewnątrz ramki opóźnienie między bajtami musi zawierać się w czasie trwania 10 bitów. Większe opóźnienie będzie traktowane jako początek nowej ramki.

**Ramka polecenia z echem**

**Ramka odpowiedzi**

### 1.2.1 Ramka polecenia

Ramka polecenia składa się z bajta adresu slave'a, bajta polecenia, bloku zawierającego od 4 do 32 bajtów danych związanych z poleceniem i bajta zawierającego sumę kontrolną. Bit 7 wartości *ADR* jest ustawiony zawsze na 0.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ADR | CMD | DAT | | | | | CHK |
| ADR | CMD | LSW | | MSW | | EXT | CHK |
| ADR | CMD | B0 | B1 | B2 | B3 | ... | CHK |

Pole *CMD* jest podzielone na dwie części. Część *CCCCCC* stanowi kod polecenia.

| CMD | | | | | | | |
|---|---|---|---|---|---|---|---|
| E | E | C | C | C | C | C | C |

Część *EE* wyznacza rozmiar bloku danych zgodnie ze wzorem

$$DATA\_LEN = 4 \bullet 2^{EE} = 4 \ll EE.$$

Gdzie EE jest rozmiarem bloku danych i może odpowiednio wynosić 4, 8, 16 lub 32 bajty.

Bajt *CHK* jest sumą kontrolną wyliczoną jako XOR z reszty danych z ramki:

$$CHK = ADR \oplus CMD \oplus DAT[i],\ i = 0 .. DATA\_LEN - 1.$$

### 1.2.2 Ramka odpowiedzi

Ramka odpowiedzi jest zbudowana z dwóch bajtów stanu, sześciu bajtów danych i jednego bajta z sumą kontrolną.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| STA | | DAT | | | | TAG | | CHK |
| STA | | LSW | | MSW | | TAG | | CHK |
| S0 | S1 | B0 | B1 | B2 | B3 | T0 | T1 | CHK |

Pole *STA* zawiera flagi stanu modułu, typu polecenia odrzuconego, alarmu, zajętości, trybu aktualnej pracy.:

*STA*

| 15 | 14 - 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| REJECT | 0 | SIGNALED | BUSY | MODE | | | | | ALARM |

*REJECT* - polecenie zostało odrzucone

*MODE* - tryb aktualnej pracy zakodowany na 5 bitach. Zobacz polecenie MODE.

*ALARM* - moduł jest w alarmie i aktualny tryb wynosi 0 (OFF); aby wyczyścić alarm i reaktywować moduł należy wysłać polecenie MODE wraz z żądanym trybem

Dla poleceń związanych z przesunięciem, pole *DAT w ramce odpowiedzi* zawiera aktualną pozycję, a pole *TAG* w ramce odpowiedzi - stan generatora trajektorii.

*TAG* (trajektoria)

| 15 - 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 - 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | LIMN | LIMP | LIM | STALL | TRIGG | INMOTION | 0 | DONE |

*INMOTION* - aktualna prędkość jest różna od 0

*DONE* - cel generatora trajektorii został osiągnięty

Pole *CHK* jest obliczane tak jak dla ramki polecenia.

## 2. Polecenia

**Opis poleceń**

| EE | CC | Polecenie | Opis | CAT |
|---|---|---|---|---|
| 0 | 01h | **PING** | Stan zapytania | SYS |
| 0 | 04h | **MODE** | Zmiana trybu pracy | SYS |
| 0 | 20h | **TRJINIT** | Inicjalizacja generatora trajektorii | TRJ |
| 0 | 21h | **START** *v* [, *a*] | Rozpoczęcie ruchu | TRJ |
| 0 | 22h | **STOP** [*d*] | Zatrzymanie | TRJ |
| 0\|1 | 23h | **GOTO** *p* [, *v*, *a*] | Przesunięcie na daną pozycję | TRJ |
| 0\|1 | 24h | **STEP** *d* [, *v*, *a*] | Przesunięcie względne | TRJ |

Polecenia trajektorii są akceptowane wyłącznie w trybie « Polecenie zdalne (Remote) », zobacz polecenie MODE.

**PING** Stan zapytania SYS

**Opis:** Polecenie pozwala masterowi sprawdzić obecność modułu i odczytać jego stan. Zwykle wywoływane co 100ms.

**Ramka polecenia:**

| ADR | 01h | 0 | 0 | 0 | 0 | CHK |
|---|---|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

**MODE** Zmiana trybu pracy SYS

**Opis:** Polecenie tymczasowo zmienia tryb pracy.
Po włączeniu tryb pracy wybrany podczas konfiguracji jest ustawiany automatycznie.

**Ramka polecenia:**

| ADR | 04h | MOD | 0 | 0 | 0 | CHK |
|---|---|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | - | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*MOD* - nowy tryb działania

| **Wartość** | **Tryb** |
|---|---|
| 0 | OFF |
| 1 | Special |
| 3 | Remote – Polecenie zdalne |
| 4 | Cursor - position |
| 5 | Speed GOP i GON |
| 6 | Speed GO i DIR |
| 7 | Speed STARTP/STOPP |

| | |
|---|---|
| 8 | Speed START/STOP i DIR |
| 9 | Travel GOP/GON |
| 10 | Travel GO/DIR |
| 11 | Push - pull |
| 13 | Push – pull with home switch |
| 14 | External A1/B1 quadrature clock |
| 15 | External A1/B1 clock & direction |
| 16 | External A2/B2/GO/DIR quadrature clock |
| 17 | External A2/B2/GO/DIR clock & direction |
| 18 | Hybrid A1/B1 quadrature clock |
| 19 | Hybrid A1/B1 clock & direction |
| 20 | Hybrid A2/B2/GO/DIR quadrature clock |
| 21 | Hybrid A2/B2/GO/DIR clock & direction |

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

**TRJINIT** Inicjalizacja generatora trajektorii TRJ

**Opis:** Polecenie pozwala zainicjalizować generator trajektorii i ustawić wartości początkowe na 0. Polecenie jest uruchamiane automatycznie przy zmianie trybu pracy. Wszelkie trwające przesunięcia zostają zaniechane.

**Ramka polecenia:**

| ADR | 20h | 0 | 0 | 0 | 0 | CHK |
|---|---|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

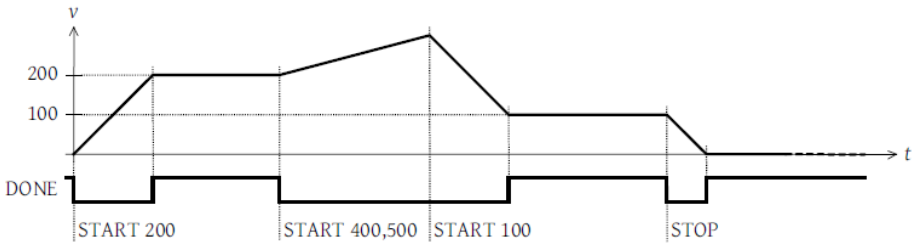
*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

| **START *vel*** | Rozpoczęcie ruchu z maksymalnym przyspieszeniem | TRJ |
|---|---|---|

**Opis:**

Generator trajektorii jest programowany do wprowadzenia silnika w prędkość *VEL*. Bit *DONE* pola *TAG* wynosi 0 podczas przyspieszania i przechodzi w 1 po osiągnięciu żądanej prędkości. Polecenie jest akceptowane nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania. Prędkości są związane ze stopniem przyspieszenia maksymalnego. Kierunek obrotu jest określany przez znak parametru *VEL*.




**Ramka polecenia:**

| ADR | 21h | $VEL_{16}$ | 0 | 0 | CHK |
|---|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*VEL* - prędkość docelowa w rpm, od -32768 do 32767

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

**START *vel, acc*** Rozpoczęcie ruchu z ustalonym przyspieszeniem TRJ

**Opis:** Generator trajektorii jest programowany do wprowadzenia silnika w prędkość *VEL*. Bit *DONE* pola *TAG* wynosi 0 podczas przyspieszania i przechodzi w 1 po osiągnięciu żądanej prędkości. Polecenie jest akceptowane nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania. Prędkości są związane ze stopniem przyspieszenia *ACC*. Kierunek obrotu jest określany przez znak parametru *VEL*.

**Ramka polecenia:**

| ADR | 21h | $VEL_{16}$ | $ACC_{16}$ | CHK |
|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*VEL* - prędkość docelowa w rpm, od -32768 do 32767

*ACC* - przyspieszenie/spowolnienie w rpm/s, od 1 do 32767, jeśli 0 - maksymalne

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

**STOP** Zatrzymanie z maksymalnym hamowaniem TRJ

**Opis:** Generator trajektorii jest programowany do zatrzymania silnika. Bit *DONE* pola *TAG* wynosi 0 podczas hamowania i przechodzi w 1 po osiągnięciu prędkości zerowej. Polecenie jest akceptowane nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania. Stopień hamowania jest określony przez parametr hamowania maksymalnego.

**Ramka polecenia:**

| ADR | 22h | 0 | 0 | 0 | 0 | CHK |
|---|---|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

**STOP *dec*** Zatrzymanie z ustalonym hamowaniem TRJ

**Opis:** Generator trajektorii jest programowany do zatrzymania silnika. Bit *DONE* pola *TAG* wynosi 0 podczas hamowania i przechodzi w 1 po osiągnięciu prędkości zerowej. Polecenie jest akceptowane nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania. Stopień hamowania jest określony parametrem *DEC*.

**Ramka polecenia:**

| ADR | 22h | 0 | 0 | $DEC_{16}$ | CHK |
|---|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*DEC* - hamowanie w rpm/s, od 0 do 32767, jeśli 0 - maksymalne

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

**GOTO *dst*** Przesunięcie do danej pozycji TRJ

**Opis:** Generator trajektorii jest programowany do umieszczenia silnika na pozycji *DST.* Bit *DONE* pola *TAG* wynosi 0 podczas przesunięcia i przechodzi w 1 po osiągnięciu pozycji docelowej. Prędkość związana jest ze stopniem przyspieszenia

maksymalnego i ograniczona prędkością maksymalną. Polecenie jest akceptowane nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania.

**Ramka polecenia:**

| ADR | 23h | $DST_{32}$ | CHK |
|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*DST* - pozycja docelowa w µkrokach, na 32 bitach ze znakiem

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

**GOTO *dst, vel, acc*** Przesunięcie do danej pozycji z określoną prędkością i przyspieszeniem TRJ

**Opis:** Generator trajektorii jest programowany do umieszczenia silnika na pozycji *DST.* Bit *DONE* pola *TAG* wynosi 0 podczas przesunięcia i przechodzi w 1 po osiągnięciu pozycji docelowej. Prędkość związana jest ze stopniem przyspieszenia ACC i ograniczeniem prędkość przez wartość *VEL*. Polecenie jest akceptowane nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania.

**Ramka polecenia:**

| ADR | 63h | $DST_{32}$ | $VEL_{16}$ | $ACC_{16}$ | CHK |
|---|---|---|---|---|---|

**Ramka odpowiedzi:**

| $STA_{16}$ | $POS_{32}$ | $TRJ_{16}$ | CHK |
|---|---|---|---|

*ADR* - adres modułu, od 0 do 127, domyślnie 1

*VEL* - ograniczenie prędkości w rpm, od 0 do 32767, max jeśli 0

*ACC* - przyspieszenie/hamowanie w rpm/s, od 0 do 32767, max jeśli 0

*DST* - pozycja docelowa w µkrokach, na 32 bitach ze znakiem

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

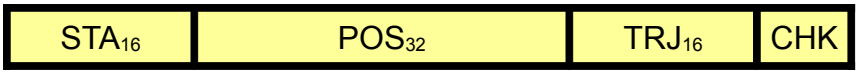
*CHK* - suma kontrolna

| **STEP *dst*** | Przesunięcie względne | TRJ |
|---|---|---|

**Opis:** Generator trajektorii jest programowany do umieszczenia silnika na pozycji aktualnej + *DST.* Bit *DONE* pola *TAG* wynosi 0 podczas przesunięcia i przechodzi w 1 po osiągnięciu pozycji docelowej. Prędkość jest związane są ze stopniem przyspieszenia maksymalnego i ograniczona przez prędkość maksymalną. Polecenie jest akceptowane nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania.

**Ramka polecenia:**


**Ramka odpowiedzi:**




*ADR* - adres modułu, od 0 do 127, domyślnie 1

*DST* - odległość w µkrokach, na 32 bitach ze znakiem

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna

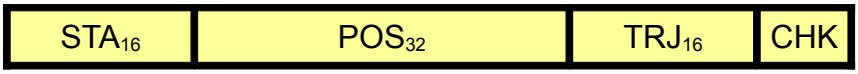
| **STEP *dst, vel, acc*** | Przesunięcie względne z określoną prędkością i przyspieszeniem | TRJ |
|---|---|---|

**Opis:** Generator trajektorii jest programowany do umieszczenia silnika na pozycji aktualnej + *DST.* Bit *DONE* pola *TAG* wynosi 0 podczas przesunięcia i przechodzi w 1 po osiągnięciu pozycji docelowej. Prędkość jest związana ze stopniem przyspieszenia ACC i ograniczona przez prędkość *VEL*. Polecenie jest akceptowane

nawet jeśli poprzednie polecenie zmiany trajektorii jest w trakcie wykonywania.

**Ramka polecenia:**


**Ramka odpowiedzi:**




*ADR* - adres modułu, od 0 do 127, domyślnie 1

*VEL* - ograniczenie prędkości w rpm, od 0 do 32767, max jeśli 0

*ACC* - przyspieszenie/hamowanie w rpm/s, od 0 do 32767, max jeśli 0

*DST* - pozycja docelowa w µkrokach, na 32 bitach ze znakiem

*POS* - aktualna pozycja w µkrokach, na 32 bitach ze znakiem

*TRJ* - stan generatora trajektorii zakodowany na 16 bitach, T0 jest najmłodszym bajtem

*STA* - stan modułu zakodowany na 16 bitach, S0 jest najmłodszym bajtem

*CHK* - suma kontrolna



**Kontakt:**
GRYFTEC
Tuwima 21
71-426 Szczecin
Polska

www.gryftec.com
**Telefon**: +48-78-46-95-491
**E-mail**: automatyka@gryftec.com