**Ukaz preko odložišča:**

Format: UkazZaStepDrive( UKAZ\_S\_PARAMETRI )

Ko se na odložišču pojavi tekst v tem formatu, se izvede ukaz. V primeru, da se pričakuje odgovor na ukaz, se odgovor doda na koncu teksta v odložišču.

**Ukaz preko ComPorta:**

Format : UKAZ\_S\_PARAMETRI

Na PC je potrebno namestiti com0com gonilnike, ki naredijo programsko simulacjo COM portov. COM porti se dodajajo v parih, komunikacija poteka znotraj vsakega para. StepDrive aplikacija izpiše ime COM porta, ki ga je potrebno uporabiti za komunikacijo.

Ko se na ustrezen COM port pošlje veljaven ukaz, se ukaz izvrši. Če se pričakuje odgovor na ukaz, se udgovor pojavi na COM Portu.

**Struktura ukaza s parametri:**

UKAZ\_S\_PARAMETRI => Ukaz ; ŠtevilkaMotorja ; ParameterUkaza

**Primer ukaza:**

*Set\_MicroStep; 1; 128* - Ta ukaz pomeni, da kontrolerju št.1 nastavi število mikro-korakov na vrednost 128.  
Preko odložišča bi isti ukaz bil: *UkazZaStepDrive(Set\_MicroStep; 1; 128 )*

**Ukazi:**

* ***LoadConfiguration\_FromFile;m;Path –> Opozorilo***
  + Naloži konfiguracijo iz datoteke, ki je predpisana s paramerom ***Path***. Ko se parametri naložijo, izbriše tudi statuse. Če datoteka ne obstaja, vrne obvestilo »Datoteka ne obstaja«.
* ***Run\_WithSpeed; m; Hitrost -> Hitrost***
  + Požene motor z ustrezno hitrostjo. Hitrost je predznačeno racionalno število, enote za hitrost so [*koraki/s*]. Koraki so polni koraki koračnega motorja. Vrne dejansko nastavljeno hitrost. Med pospeševanjem je vklopljen status »Busy«, ukaz se lahko da kadarkoli in se takoj izvede.
* ***Move\_For; m; Premik -> Premik***
  + Premakne motor za ustrezno število mikro-korakov. Premik je predznačen in celoštevilčen. Absolutni premik je odvisen od nastavitve števila mikro-korakov na korak. Med premikanjem je vklopljen status »Busy«, pred ukazom mora motor mirovati, največji možen premik je 4194303.
* ***GoTo\_Short; m; Pozicija -> Pozicija***
  + Pošlje motor s pospeškom in izbrano maksimalno hitrostjo na zahtevano pozicijo po najkrajši poti, pozicija je v mikro-korakih v območju +/- 2097151. Med premikanjem je vklopljen status »Busy«.
* ***GoTo\_DirPos; m; Pozicija -> Pozicija***
  + Pošlje motor s pospeškom in izbrano maksimalno hitrostjo na zahtevano pozicijo. Motor zavrti v pozitivni smeri, pozicija je v mikro-korakih v območju +/- 2097151. Med premikanjem je vklopljen status »Busy«, pred ukazom mora motor mirovati.
* ***GoTo\_DirNeg; m; Pozicija -> Pozicija***
  + Pošlje motor s pospeškom in izbrano maksimalno hitrostjo na zahtevano pozicijo. Motor zavrti v negativni smeri, pozicija je v mikro-korakih v območju +/- 2097151. Med premikanjem je vklopljen status »Busy«, pred ukazom mora motor mirovati.
* ***Stop\_Hard; m***
  + Ustavi motor takoj. Ko se motor ustavi, pusti izhodno stopnjo prižgano.
* ***Stop\_Soft; m***
  + Ustavi motor s prednastavljenim pojemkom. Ko se motor ustavi, pusti izhodno stopnjo prižgano.
* ***DriverHiZ; m***
  + Izklopi izhodno stopnjo, posledično se ustavi vrtenje.
* ***GetClear\_Status; m -> Status***
  + Prebere statusni register in vrne njegovo vrednost. Po branju izbriše statuse, Interpretacija statusnega registra je v podatkovnem listu.
* ***Get\_Status; m -> Status***
  + Prebere statusni register in vrne njegovo vrednost. Interpretacija statusnega registra je v podatkovnem listu.
* ***Set\_Position; m; Pozicija -> Pozicija***
  + Nastavi števec pozicije na željeno vrednost, vrne nastavljeno vrednost. Ukaz se izvede, če se motor ne premika.
* ***Set\_MarkPosition; m; Mark -> Mark***
  + Nastavi »Mark« register na željeno vrednost, vrne nastavljeno vrednost. Ukaz se vedno izvede.
* ***Set\_Acceleration; m; Pospešek -> Pospešek***
  + Nastavi pospešek na željeno vrednost, enote so [korak/s2]. Vrne nastavljeno vrednost.
* ***Set\_Deceleration; m; Pojemek -> Pojemek***
  + Nastavi pojemek na željeno vrednost, enote so [korak/s2]. Vrne nastavljeno vrednost.
* ***Set\_MaxSpeed; m; Hitrost -> Hitrost***
  + Nastavi vrednost največje hitrosti, enota je [korak/s], vrne nastavljeno vrednost.
* ***Set\_MinSpeed; m; Hitrost -> Hitrost***
  + Nastavi vrednost najmanjše hitrosti, enota je [korak/s], vrne nastavljeno vrednost.
* ***Set\_HoldCurrent; m; Tok -> Tok***
  + Nastavi tok, ki naj teče med mirovanjem, enote so [mA], vrne nastavljeno vrednost.
* ***Set\_RunCurrent; m, Tok -> Tok***
  + Nastavi tok, ki naj teče med premikanjem s stalno hitrostjo, enote so [mA], vrne nastavljeno vrednost.
* ***Set\_AccCurrent; m; Tok -> Tok***
  + Nastavi tok, ki naj teče med pospeševanjem, enote so [mA], vrne nastavljeno vrednost.
* ***Set\_DecCurrent; m; Tok -> Tok***
  + Nastavi tok, ki naj teče med zmanjševanjem hitrosti, enote so [mA], vrne nastavljeno vrednost.
* ***Set\_MicroStep; m; µKorak -> µKorak***
  + Nastavi koliko mikro-korakov naj bo na korak. Vrne nastavljeno vrednost. Vrednost se lahko nastavlja samo, ko je izklopljena izhodna stopnja.
* ***Get\_IsBusy; m -> Status***
  + Če je vklopljen status »Busy«, vrne 1, drugače vrne 0.
* ***Get\_IsBridgeOn; m -> Status***
  + Če je izhodna stopnja vklopljena, vrne 1, drugače vrne 0.
* ***Get\_IsCommandError; m -> Status***
  + Prebere »CommandError« status. Če je bil Error, vrne 1 drugače vrne 0.
* ***Get\_IsSwitchOn; m -> Status***
  + Pogleda, če je končno stikalo sklenjeno. Če je sklenjeno, vrne 1, drugače vrne 0.
* Get\_Acceleration; m
* Get\_Deceleration; m
* Get\_MaxSpeed; m
* Get\_MinSpeed; m
* Get\_Position; m
* Get\_MarkPosition; m
* Get\_Speed; m
* Get\_HoldCurrent; m
* Get\_RunCurrent; m
* Get\_AccCurrent; m
* Get\_DecCurrent; m
* Get\_MicroStep; m
* GoToSwitch\_ResetAbs\_SoftStop; m // Motion with desired speed until SW closed, after event SoftStop. At SW event ABS register is reset
* GoToSwitch\_AbsToMark\_SoftStop ; m // Motion with desired speed until SW closed, after event SoftStop. At SW event MARK register is preset with ABS register
* ReleaseSwitch\_ResetAbs; m // Motion with minimum speed until SW opened, after event HardStop. At SW event ABS register is reset
* ReleaseSwitch\_AbsToMark ; m // Motion with minimum speed until SW opened, after event HardStop. At SW event MARK register is preset with ABS register