Help sheet 19

ST Robotics

Continuous read-back from the DSP.

Examples follow. Use the Acrobat select tool to copy text into your project.

METHOD 1

This method applies if you initiate a move with DSPMOVE DSPSMOOTH or CRUN.
With these modes the CPU sends your coordinates to the DSP and returns to command prompt (or continues with the next word in your definition). The DSP is controlling robot motion without the involvement of the CPU.

NOTE: in normal use the CPU enters a loop waiting for the DSP to finish. In this loop the CPU monitors the stop button. In this mode pressing the stop button will not stop the robot because the CPU is not checking. You need to make your own checks and send STOP to the DSP.

The following examples are for DSPSMOOTH

Here are 2 ways to send coordinates to the CPU thence to the DSP:

```
: JMO ( absolute move
PREP
NEWTARGET
MOTORS 0 DO
 TARGET I 2* + !
LOOP
TARGET DSPSMOOTH DROP
;
```

```
: JMOR ( relative move
PREP
NEWTARGET
MOTORS 0 DO
 I GLOBALS @ +
 TARGET I 2* + !
LOOP
TARGET DSPSMOOTH DROP
;
```

How to monitor the stop button

Make a loop and include the word

```
?STOP
```

It will exit when the DSP has finished it's move or if you press the stop button.

Continued...

```
6 USARRAY TEMPS
: PREP
CR
CLRDSP
MOTORS 0 DO
  I GLOBALS @ I TEMPS !
LOOP
SET DSPVEC UPDATE
;

: UPDATE ( like DSPASSUME but can be used more than once
18 SPSB
0 DSPCHANS @ 1- DO
  I TEMPS @ I GLOBALS !
  SPRW I GLOBALS +!
-1 +LOOP
CFLAG C0SET
( the following line is only needed if you want to read back Cartesian
( position. Note that the final declared position may well be 0.1
( different from the target position because of forward kinetics errors.
COMPUTE
;
```

Each command must be preceded by the word PREP. The joint count and Cartesian position variables are then updated all through the motion. To disable this mode use PREPOFF
Thereafter use UPDATE as often as you need. Follow with WHERE or query individual axes e.g.
`X ? Y ? Z ?` etc.

The DSP only works with motor counts. After each use of UPDATE the GLOBAL variables are updated. You can read them back with WHERE. To read the Cartesian position use COMPUTE as below. Note that there is often an error of 0.1 due to RoboForth forward kinetics errors.

METHOD 2
Using vectored execution

This method only works with the following commands:
-GOTO (not GOTO only -GOTO)
Cartesian MOVE
Cartesian MOVETO
RobWin drop down Cartesian absolute and relative move
RUN in continuous mode only.

How to read back the DSP
There is a word DSPASSUME that reads back the DSP after it has stopped but is not suitable for use while the robot is still running. So here is a new word UPDATE that refers back to the position the robot was in at the start of the move (in the array TEMPS)

Each command must be preceded by the word PREP. The joint count and Cartesian position variables are then updated all through the motion. To disable this mode use PREPOFF

The DSP only works with motor counts. After each use of UPDATE the GLOBAL variables are updated. You can read them back with WHERE. To read the Cartesian position use COMPUTE as below.

```
6 USARRAY TEMPS
: UPDATE ( like DSPASSUME but can be used more than once
10 MSECS ( this determines how often the position is updated
18 SPSB
0 DSPCHANS @ 1- DO
  I TEMPS @ I GLOBALS !
  SPRW I GLOBALS +!
-1 +LOOP
CFLAG C0SET
( the following line is only needed if you want to read back Cartesian
( position. Note that the final declared position may well be 0.1
( different from the target position because of forward kinetics errors.
COMPUTE
( the following line is just for testing and continually prints on screen
( the current values of X Y and Z while the robot moves. Replace with
( whatever you need to get at the values
13 EMIT X ? Y ? Z ? 5 SPACES
;
: PREP
CR
CLRDSP
MOTORS 0 DO
  I GLOBALS @ I TEMPS !
LOOP
SET DSPVEC UPDATE
;
```

```
: PREPOFF
0 DSPVEC !
;
: HOME
PREPOFF
HOME
;
```

For your convenience the above motion commands can be redefined so you don't have to keep typing PREP before each command.

```
: MOVETO
PREP MOVETO PREPOFF
;
: MOVE
PREP MOVE PREPOFF
;
: CM
PREP CM PREPOFF
;
: RUN
1 GOTO
CONTINUOUS
PREP
RUN
PREPOFF
;
```