

Absolute Calibration

A technique for calibrating the robot to an absolute position in the work space and thus improve on the repeatability of the internal homing sensors.

See this clip

<http://www.youtube.com/watch?v=CMdurikIE0U>

The clip shows the absolute calibration followed by checking the calibration against an electronic dial-gauge (LVDT). The LVDT will not be described here. You can use a standard dial gauge to check that the calibration is working.

You need a 8mm PNP open collector proximity detector such as may be obtained from Balluff.

Mount this on a bracket as shown in the clip.

Connect the sensor to 9-way D connector PB 7 (see controller manual for pinouts)

Check it works with a piece of steel. Enter PP and offer the steel to the sensor.

11111110

steel on

01111110

steel off

11111110

press esc to exit.

Mount a square of steel (reflector) on the robot hand such that the robot can offer the square to the detector.

The detector should be at X=0 so that any change in Y does not result in any waist motion.

Enter the code below into a new project or into your existing project.

Enter READY for an R12. For an R17 get the robot into CARTESIAN mode and position it so the reflector is flat i.e. level with the X-Y plane.

Use the JOG function to get within 5-10mm of the sensor. Do NOT use X motion.

Enter WHERE and note the Cartesian position. Enter those values for PROX in the code below.

(values shown are for the clip – yours will be different)

DECIMAL

: VSLOW 250 SPEED ! 250 ACCEL ! ;

: SLOW 400 SPEED ! 400 ACCEL ! ;

: FAST 10000 SPEED ! 2000 ACCEL ! ;

: PROX

0 X ! 3620 Y ! -1720 Z ! 900 PITCH ! 0 W !

TRANSFORM -GOTO

;

Help sheet 16



0SENSE stops motion when the sensor goes to 0, 1SENSE stops motion when the sensor goes to 1

```
: 0SENSE
BEGIN
PB 7 BIT? 0=
?RUN 0=
OR UNTIL
```

```
;
: 1SENSE
BEGIN
PB 7 BIT? 0 >
?RUN 0=
OR UNTIL
```

```
;
ZSENSE moves the robot down slowly until the sensor changes to 0.
```

```
: ZSENSE
SLOW
-50 Z +!           ( AIM FOR 5MM LOWER
TRANSFORM DSPSMOOTH ( TELL DSP TO DO IT
DROP              ( DROP THE UNWANTED TRANSFORM RESULT
0SENSE           ( MOVE UNTIL PB 7 GOES 0
STOP             ( STOP THE DSP
DSPASSUME        ( ASK THE DSP WHERE IT GOT TO
COMPUTE          ( UPDATE THE CARTESIAN POSITION
;
```

YSENSE moves the robot back on Y axis only until the sensor changes to 1

```
: YSENSE
SLOW
-100 Y +!
TRANSFORM DSPSMOOTH DROP
1SENSE
STOP
DSPASSUME
COMPUTE
;
```

XSENSE moves in +X but only the waist is moved. That way the other axes do not move to confuse the calibration or create vibration.

```
: XSENSE
TELL WAIST 200 DSPMOVE
1SENSE
STOP
DSPASSUME
COMPUTE
;
```

Help sheet 16



3SENSE does all 3

```
: 3SENSE
FAST          ( GO FAST TO PROX
PROX
VSLOW
ZSENSE        ( FIND Z FIRST
0 0 -10 MOVE  ( MOVE DOWN A BIT MORE OR THE YSENSE MIGHT BE PREMATURE
500 MSECS     ( SETTLING TIME
YSENSE        ( THEN BACK ON Y – MULTI-AXIS
0 20 0 MOVE
500 MSECS
XSENSE        ( WAIST ONLY
JOINT
;
```

Use 3SENSE one time to find out where it ends up. Enter WHERE and enter the values in NEST

```
CREATE NEST
142 , 7360 , 4960 , 1181 , 1328 , 0 , 0 , 0 ,
```

Finally use ABSCAL to restore those values each time.

```
: ABSCAL
FAST
HOME CALIBRATE
READY
PROX
3SENSE
NEST ASSUME ENCSET
FAST
READY
;
```