

Getting started with ROBWIN and ROBOFORTH II v10-12

RobWin © Sands Technology International 1996-2006; This is a Windows based application which runs in the computer. It communicates with the robot controller and also saves robot software on disk. It keeps an identical copy of the controller data in memory while you use it.

Roboforth II © D. N. Sands 1984-2008. This is a complete system which resides in the robot controller. It is based on the FORTH control language. It is extensive but not greatly complex. As with any software the best way to learn it is to use it. This document walks you through basic motions and creation of simple tasks.

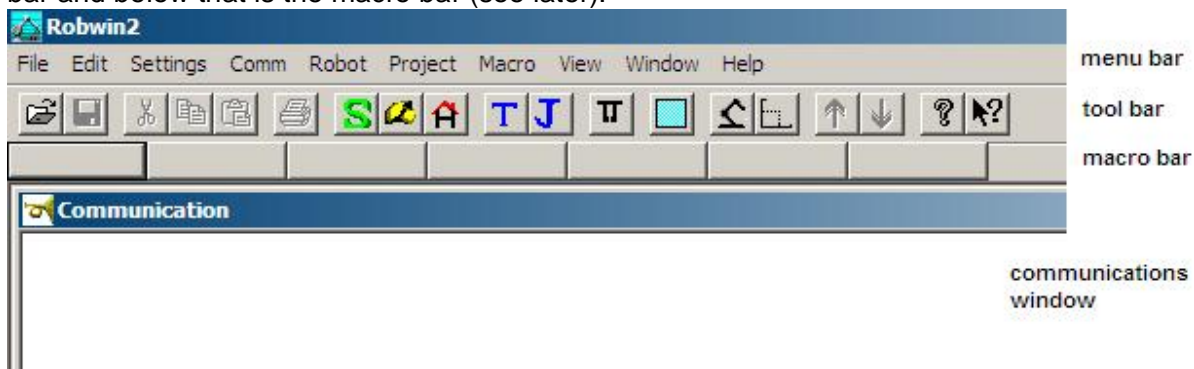
Note: Only type what is underlined. When you've typed it press the enter key.

Programming the robot requires communication with the robot controller. When ROBWIN is running there is a communication window. You can type things in this window and they are sent directly to the controller i.e. this window is like a terminal. ROBWIN also sends commands to the controller, most of which appear in the communication window. ROBWIN also has the ability to download data to the controller and upload data from it, and can save or reload data from disk.

Once you are in ROBWIN you can click on Help to get an explanation of any command or variable which may be sent by ROBWIN. For all other ROBOFORTH commands use the glossary or the manual. You can also click on the help button which is an arrow plus a question mark. A large question mark appears by the cursor which you then take to a tool bar button or any menu command for an explanation of that function.

Launch

Double-click on the ROBWIN icon. Along the top are the menu buttons, below that is the robot tool bar and below that is the macro bar (see later).



Load/check settings: On the menu bar click settings, open file. If you have an R17 click on R17v10.CFG; if an R19 click on R19v10.CFG, if an R15 click on R15.CFG. Values may be manually set using settings, configuration – **make sure DSP, bank memory and hide mode are checked.**

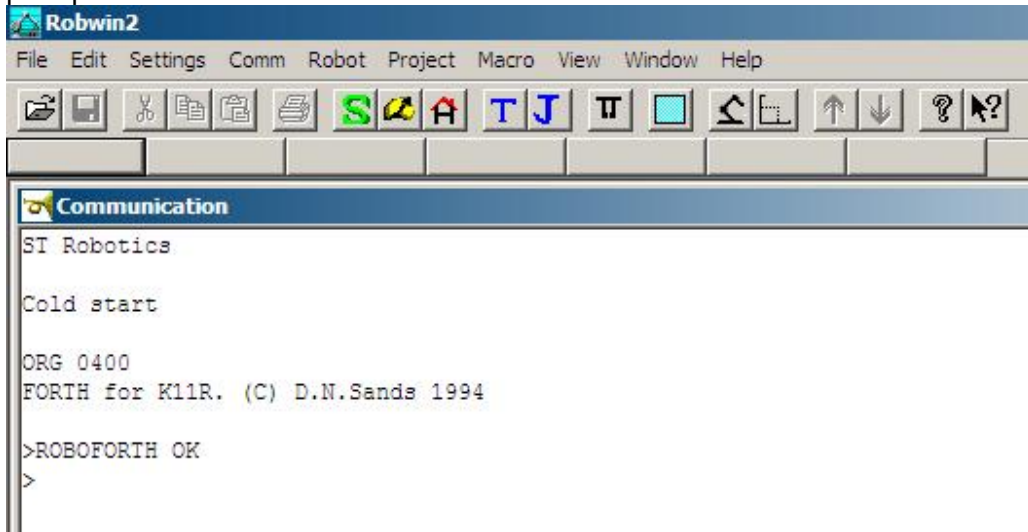
caps lock – All ROBOFORTH commands are in upper case, ensure CAPS LOCK is on.

Switch the key switch to cold start and switch on the robot controller.


In the communications window (the console) you should see a herald which includes the words 'cold start' (see next page) and a chevron prompt >.


Getting started, a primer


Type ROBOFORTH into the console. The response should be OK and a new line with a chevron prompt >



Note: When you type anything in the console all buttons are greyed out. If you back off what you typed they will still be greyed. Press **esc** if in doubt.

In the tool bar press the start button  or type START into the communications window. This energizes motors, zeros all position counts, sets default values for SPEED etc.

Press the Calibrate button  or type CALIBRATE Robot moves to the calibrate position and calibrates all axes.

Press the Home button  or type HOME Robot moves to the HOME position in which all motor counts are zero.

Basics of FORTH

Forth is a language comprised of 'words' in a dictionary. Each word is a definition of a procedure defined in terms of words further down the dictionary. A word may be spelled with any characters on your keyboard, for example HOME is a word and + is a word.

Values or 'arguments' that the words need are put on a stack for the word to use up. For example 1000 MOVE moves a selected joint 1000 steps. If you wanted to add 4 and 5 and print the result you would enter

```
4 5 + .
```

the value 4 is put on the stack, then the 5. Then the word + picks them both off the stack, adds them together and leaves the result (9) on the stack. The word . (dot) which is a Forth shortform for print then takes the number off the stack and displays it on the screen.

Teaching the robot is in 2 stages:

First **move** the robot to the desired position, second **learn** that position.

MOVEMENT

Try these commands in the communication window:

RELATIVE COMMANDS

TELL WAIST 1000 MOVE Moves waist 1000 motor counts.

The response is OK when the motion is complete.

1000 MOVE Waist still selected, moves another 1000

-3000 MOVE Waist moves back 3000 counts.

Note that the value (argument) comes *before* the command not after.

WHERE Displays current position of the robot.

TELL SHOULDER 5000 MOVE Moves R17 shoulder 5000 motor counts.

ELBOW 2000 MOVE (without TELL) adds elbow to selection both move 2000

TELL EXTEND 5000 MOVE Moves the R19 extend axis out 5000 counts

TELL EXTEND REVERSE 2000 MOVE Moves the R19 extend axis in 2000 counts

TELL LIFT 5000 MOVE Moves the R19 lift axis up 5000 counts.

Other joint names are:

HAND selects wrist pitch (up/down movement)

WRIST selects wrist roll (rotates wrist)

ALL selects all the joints at once.

ALL 500 MOVE moves all the joints 500 motor counts.

5000 TWIST rotates the wrist 5000 motor counts. Does not change the previous selection of joints.

ABSOLUTE COMMANDS

TELL WAIST 2500 MOVETO Moves the waist until its total count is 2500.

WHERE Redisplay position of robot to show this.

UNTWIST Rotates the wrist back to its zero count position. (Same as TELL WRIST 0 MOVETO)

You can do similar things with dialog boxes:

Click Robot then Joint Relative Move for the MOVE command or Joint Absolute Move for the MOVETO equivalent. These dialog boxes allow different values for each joint.

OTHER COMMANDS

HOME Moves all the joints to their zero count positions, collectively.

or click on the home button 

GRIP Closes the gripper.

UNGRIP Opens the gripper

or click  which toggles grip open/closed

Getting started, a primer

CONTROLLING SPEED AND ACCELERATION

There are two variables SPEED and ACCEL which control speed and acceleration.

In the communications window

Type `SPEED ?` The controller tells what is current speed factor. Range: 100-32000

Type `5000 SPEED !` Sets the arm speed to factor 5000.

`SPEED ?` Confirms new speed factor.

Try it with, for example:

`TELL WAIST 4000 MOVE 0 MOVETO`

The acceleration is set by the value of ACCEL. Range 100-5000

`1000 ACCEL !` Sets acceleration to 1000

`ACCEL ?` Reports acceleration.

Try above motion with different values of ACCEL

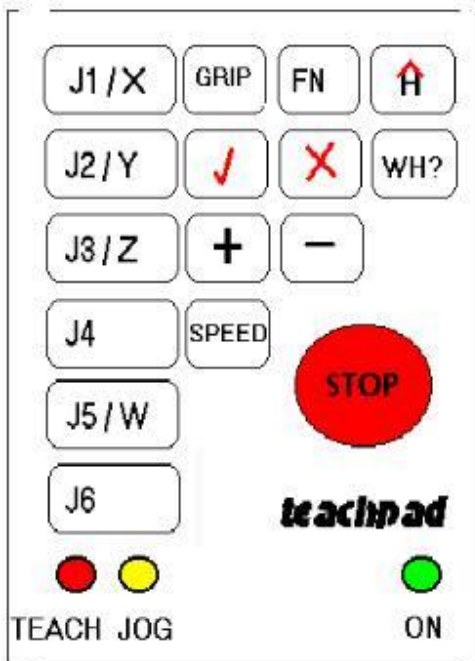
NOTE: SPEED must not be more than 16 times ACCEL.

Or click Settings, Variables

This is better because then ROBIN knows what values are in the controller.

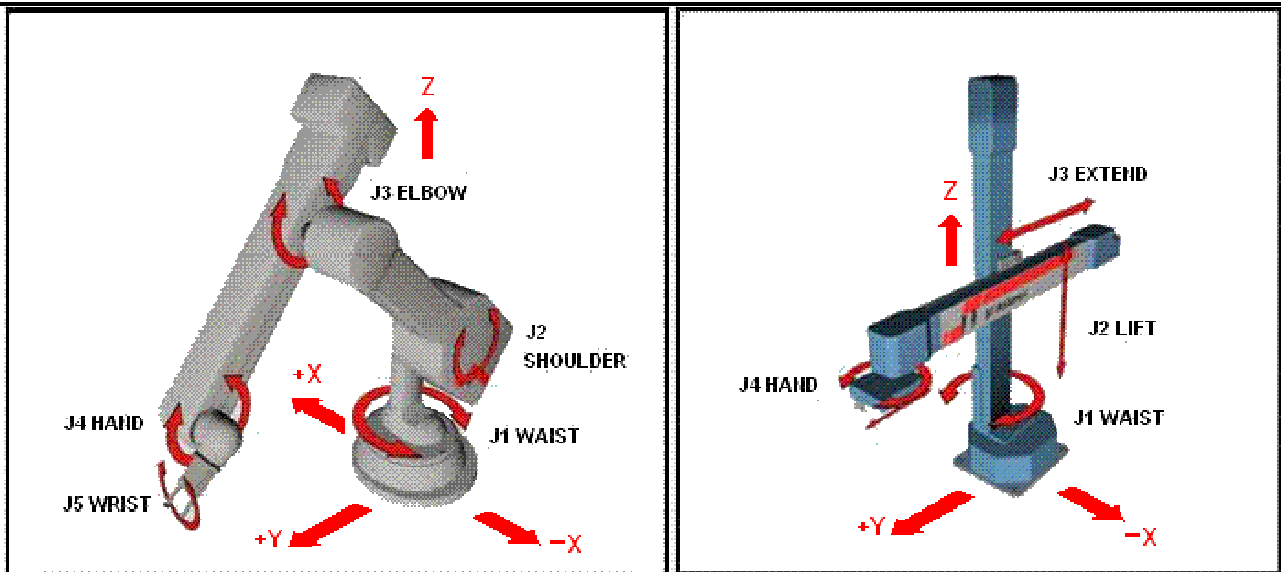
MOVING ROBOT WITH THE TEACH PAD

Press the **T** button. (red teach LED lights – green already on) The computer asks for a teach speed which defaults to 16. Don't type or click anything, pad is ready to use.



Press J1 (or J2,3,4,5) to select which joint is to move

Getting started, a primer



On R17 J1 is waist, J2 is shoulder, J3 elbow, J4 hand (pitch), J5 wrist, J6 is track.

On R19 J1 is waist, J2 is lift, J3 is extend, J4 is hand (yaw), J5 is track.

On R15 J1 is track (X), J2 is extend (Y), J3 is lift (Z), J4 is pitch, J5 is roll.

Press + and hold Moves the selected joint forwards.

Release the + key Stops movement.

Press – and hold Moves the selected joint in reverse.

Release Stops movement.

To increase the speed press SPEED then +; to decrease press SPEED then – . This is confirmed on screen.

Press the joint key again then + or – to move at the new speed.

Press the WH? key Displays current position of the robot. (same as WHERE)

Press the H key Moves all the joints to their zero positions, Same as HOME.

Press GRIP Selects the gripper.

Press [+] Closes the gripper.


Press [-] Opens the gripper.


The 0 and X are learn and delete - DO NOT PRESS AT THIS TIME.

Press the [esc] key on computer to exit teach pad.

Getting started, a primer

CARTESIAN COORDINATES

Type HOME or click  button to make sure the robot is at HOME

Now click the Cartesian button . Command sent to controller is CARTESIAN but ROBWIN also knows about it. Now on the menu bar click **Robot Cartesian position**. This displays the Cartesian position of the robot at home. Or you could have typed WHERE in the communication window which reports the current position of the robot in Cartesian coordinates. There are 6 possible values:

X Distance along X axis from the origin in mm.

Y Distance along Y axis from the origin in mm.

Z Distance along Z axis from the origin in mm.

Wrist Pitch: The angle of the hand to horizontal. Controller variable PITCH

Wrist Rotation (roll): The roll angle of the wrist. Controller variable W

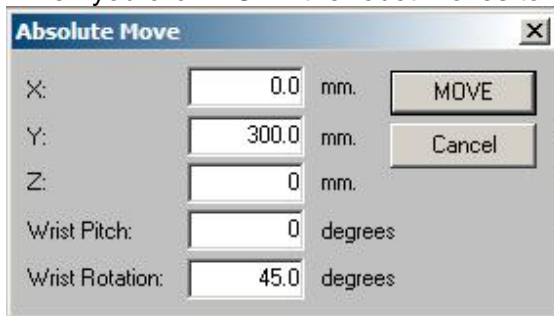
There is also another variable, TOOL-LENGTH not accessible by ROBWIN except through the communications window - see main manual later for full explanation.

On an R17 the origin (0,0,0) is at the intersection of the shoulder and waist axes. The X-Y-Z position described is the position of the intersection of the wrist rotate and hand (wrist pitch) axes.

Click **robot, Cartesian abs. Move**

This displays the current position of the robot and permits you to enter new values as required.

When you click MOVE the robot moves to that position. For example



The screenshot shows a dialog box titled "Absolute Move" with a close button (X) in the top right corner. It contains five input fields with corresponding units and two buttons: "MOVE" and "Cancel".

Parameter	Value	Unit
X:	0.0	mm.
Y:	300.0	mm.
Z:	0	mm.
Wrist Pitch:	0	degrees
Wrist Rotation:	45.0	degrees

This will move the robot to zero on the X axis, 300mm from center on the Y axis and zero on the Z axis (which for R17 means level with the shoulder pivot). Wrist rotates to 45 degrees.


Getting started, a primer

CARTESIAN COMMANDS

Alternatively in the **communications window** you can use MOVETO and MOVE which in CARTESIAN mode now require 3 values (arguments). MOVE is still relative and MOVETO is still absolute.

<u>0 400.0 0 MOVETO</u>	Moves the robot to X=0, Y=400.0 mm, Z=0.
<u>0 4000 0 MOVETO</u>	Robot does not move because 4000 is the same as 400.0
<u>0 100.0 0 MOVE</u>	Robot moves additional 100.0mm
<u>WHERE</u>	Display new position of robot. PREV line indicates previous position.
<u>TOOL</u>	This is the command to change the variables PITCH, W and TOOL-LENGTH
WRIST PITCH? enter 0	To move hand to horizontal.
WRIST ROLL? enter 90.0	To rotate wrist by 90.0 degrees (resolution of angular values is 0.1 degrees)
TOOL-LENGTH just press enter.	
ALIGN? just press enter.	
EXECUTE? press Y	Hand immediately moves to horizontal Wrist rotates 90 degrees.

MOVING ROBOT WITH THE TEACH PAD IN CARTESIAN MODE

Press the  button (**Jog**). (yellow jog LED lights – green already on) The computer asks for an increment which defaults to 10mm. Don't type or click anything, pad is ready to use.

(You can enter a new increment first if you wish but keep the default for now. The largest value you can enter is 25.5 and remember that 25.4 is one inch.)

On the keypad select X Y or Z.

Then press + or – one time (not press and hold as before). On each press the robot moves in that axis by the set increment. To increase the increment press SPEED then +; to decrease press SPEED then – Then press the X Y or Z key again and + or – to move a different axis.

J4 selects the hand – an increment of 10.0 means 10 degrees.

J5 selects wrist – an increment of 10.0 means 10 degrees.

It is best to use the dialog box first to get the robot roughly where you want it before using JOG.

TEACHING THE ROBOT

Open a project

Click **project, new** and choose a name for your project for example **myprog**.

When the project is open you will see three more windows: routes, places and a window labeled myprog.ed2

We advise (from experience) to move the communications window to the bottom and to re-arrange the other three so all are visible.

The system sends commands ROBOFORTH and STARTOVER to the controller which you can see in the communications window. This clears out the controller ready for the new session.

Teaching the robot is a process of moving to a position and learning that position. There are two devices for learning robot positions:

a ROUTE - this is a named list of positions which the robot follows in sequence.

a PLACE - this is a named single position to which the robot will go when required.

The user must choose which is most suitable for the task and must also choose a name which describes what the route or place means in the context of the application.

A task can involve more than one route or place or a mixture. All the routes and places will be saved together by ROBWIN in 2 files regardless of how many routes or places you have.

PLACES

Move the robot to any position you like using any of the methods described above.

Click the Places dialog box. If you can't see it click **Project > Places** This brings up list of existing places (empty).

Click **Add New**, enter a name for the place, for example JIG and click OK or press enter.

If you moved the robot using Cartesian coordinates the values learned in the controller and in ROBWIN will be the joint counts for that position.

To make the robot move to this position click the communications window and enter the name of the place, for example suppose the place name were JIG. Enter:

HOME robot goes to home position

JIG robot moves back to the place position.

If you regard this position as a target position you can now move the robot back to a position which is safe for the robot to move to before it goes into the target position. You then click on **Set Apprch** in the Places window. The PLACE now has two positions: the approach position and the target position. Example of using this feature

HOME Moves all the joints to zero count positions.

JIG Moves robot to JIG *via* the approach position.

WITHDRAW Moves robot back to the approach position.

Only ever use the withdraw command after the place name has been executed.

Note: just creating the place was enough to record it's position. Just entering the place name is enough to move the robot there. A Place is self learning and self executing.

If you want to change the coordinates of a place move the robot to the new position and click on the **Set to here** button.

Getting started, a primer

Example pick and place program

Learn a place, called, say JIG with an approach position.


Learn a second place, called say FEED with an approach position.

Select (click) the .ED2 window and write in the following program:


```

: TEST
FEED
GRIP
WI THDRAW
JIG
UNGRI P
WI THDRAW
HOME
;

```

Click the down arrow  to download to the controller.

Enter TEST to try it If you need to change it click the ed2 window again and when you have edited

the text click  again to download to the controller.

ROUTES

Find the **routes** window or click **Project Routes**.

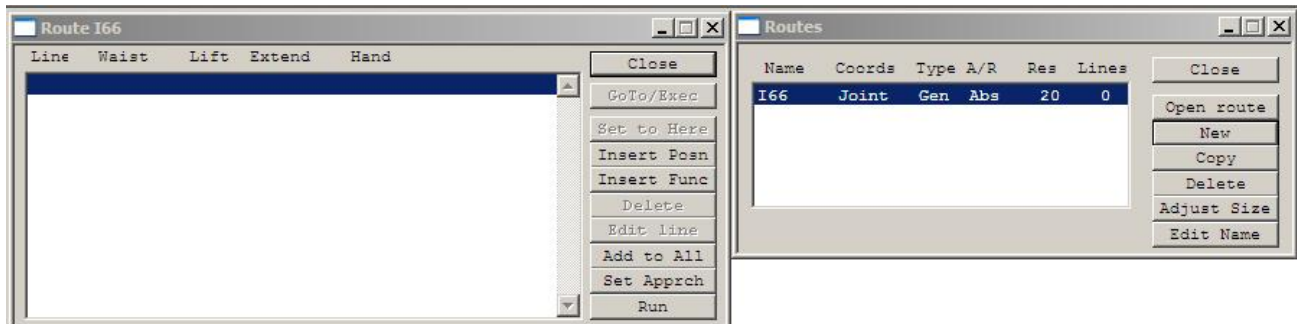
Click the **New** button

Enter a name, for example I66



You must then reserve some space for the route. The default is 20 but you can reserve hundreds if you think you'll need them.

Click OK and this creates a window for the new route:



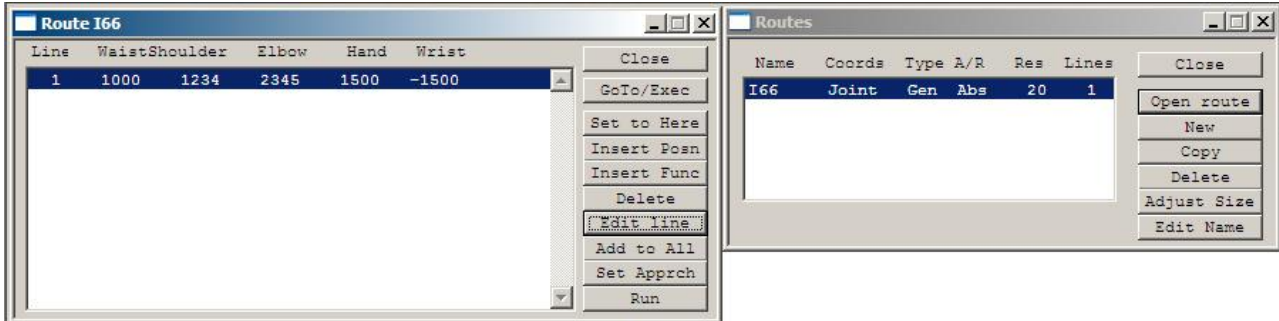
Getting started, a primer



Learning

Move the robot to desired position using any method described above. If using the teach pad press escape, don't press the tick key.

Click **Insert Posn**



For the next position highlight the next blank line and click **Insert Posn** again. Don't forget to click back inside the communications window before you type a command.

To RUN a route type RUN in the console (communication window). The robot will run from point to point stopping at each learned point.

To keep the robot running through all the points without stopping enter CONTINUOUS then RUN. (on same line or one line at a time, it makes no difference)

If you see 'too tight line....' then type ADJUST then RUN again. A new speed is announced.

The opposite of CONTINUOUS mode is SEGMENTED.

Editing a route

To change the data in a line move robot to new position, highlight the line to change and click **Set to Here**

To insert a line move the robot to desired position, highlight the line and click **Insert Posn**. The new line will be inserted in front of the highlighted line.

You can also edit the actual data in a line with **Edit Line**




You can also insert or learn a function with **Insert Func**. The function could be, for example GRIP or UNGRIP. These commands will then be carried out as the robot runs through the route.


The Insert Func also asks for a value. GRIP and UNGRIP require no value but you could, for example, change speed by inserting function SPEED then a value. The DSP speed will then change when the robot gets to that line. It will stay changed until set to another value. Another possible function is MSECS with a value, for example MSECS then 1200 inserts a 1.2 second delay.

You can only run a route with these functions in segmented mode (see manual).

Getting started, a primer

USING THE TEACH BOX TO LEARN A ROUTE

Click on  button or enter TEACH to activate the teach pad, as described above. When satisfied with the position press the red  key. This learns the position into the currently selected route. The red  deletes the last line learned.

You may of course have many routes and places. A place is self-learning (adopts the current position of the robot when it is created) and self executing (just type its name to make the robot go there). When you have more than one route you need to type its name to make other commands relevant, for example RUN (e.g. I66 RUN, PATH2 RUN etc). Likewise the tick  on the teach box learns a new position into the selected route (e.g. I66 then TEACH).

SAVING ON DISK

Click Project Save.

The computer maintains 3 files, types RUN, ED1 and ED2 for example:

myprog.run - contains all the positional data in binary.

myprog.ed1 - contains a list of the route and place names in the same order as they were created.


DO NOT EDIT THIS FILE

Also a third file myprog.ed2

CREATING A TASK WITH BOTH ROUTES AND PLACES

Click on the ED2 window. You can now enter definitions in this window, for example

```
: TASK  
JIG  
I66 RUN  
:
```

To download this file to the controller click on the  button. This also automatically saves the file to disk.

To test this function simply type

```
TASK
```

A variation might be:

```
: TASK  
10000 SPEED !  
JIG  
20000 SPEED !  
I66 CONTINUOUS ADJUST RUN  
10000 SPEED !  
:
```

Each time you click the download button the new text is automatically saved however you need to click **project, save** to save the data on disk. The text is saved in the third file, myprog.ed2.

You can also save the project in the controller Flash EPROM using the command

```
USAVE
```

If you switch off and on the controller with the front panel key in the warm start position the data and programming are restored from Flash EPROM. It is still necessary to use START and CALIBRATE.

You are now familiar with the basics of ROBOFORTH and ROBWIN. For more advanced programming please refer to the software reference manual.