

Education Cell

Technical Documentation for the R-30iB Controller

V3.3



1 Contents

1	Selecting and running Example program	4
1.1	Start Up Screen	4
1.2	Example Program "AAA_DEMO"	5
1.3	AAA_DEMO Program Sequence with / without iRVision	9
1.4	Operation Menu Display	10
1.5	4D Graphics (Option)	11
2	Demonstration Program Setup / Teaching	12
2.1	Gripper Setup.....	12
2.2	UFrame 2 Setup.....	13
2.3	Orient Origin Point.....	15
2.4	X-Direction Point	16
2.5	Y-Direction Point	17
3	Robot Setup	18
3.1	UTOOL / TCP Setup	18
3.2	Payload Setup	18
3.3	DCS Setup	19
3.3.1	DCS Tool Model	19
3.3.2	DCS Safe Zone	20
3.4	I/O Setup	21
3.5	Macro Setup.....	22
3.6	TP Function Key Hint Screen	23
3.7	Home / Reference Position	24
3.8	Power Up Program.....	25
4	iRVision Setup (Option).....	26
4.1	Camera Adjustment	26
4.2	Application Frame	28
4.3	Camera Calibration	29
4.4	Example iRVision Application	31
4.5	iRVision Tips	32
4.6	TP Program.....	34
5	List of Registers	35
5.1	Numeric registers	35
5.2	Position Registers	37
6	Program Details	38
6.1	List of programs	38
6.2	Sample Program Listings	41
7	User Interface Setup	43
7.1	Menu Utility Setup	43
7.2	HTML Screens	44
7.3	HTML Listing Extract.....	45
8	Technical Data	49

FANUC

Disclaimer

This documentation has been provided by FANUC Europe Corporation S.A. ("FANUC") for convenience and general information purposes only, with no intention of providing comprehensive, complete, accurate or up to date information. Therefore, FANUC does not warrant or assume any legal liability or responsibility whatsoever for the accuracy, completeness, or usefulness of any information, application, apparatus, product, use, processes or services disclosed herein.

Should you be interested in obtaining specific information or advice on FANUC applications, apparatus, products, use, processes or services, please contact FANUC at:

FANUC Europe Corporation
Société Anonyme

Zone Industrielle

L-6468 Echternach

Grand Duchy of Luxembourg

Tel.: +352-727777-1

INTENDED USE OF EDUCATION CELL

The Education Cell is intended for Education and Training Purposes

**It is designed for lightness and portability,
not for high speed robot motion or full acceleration.**

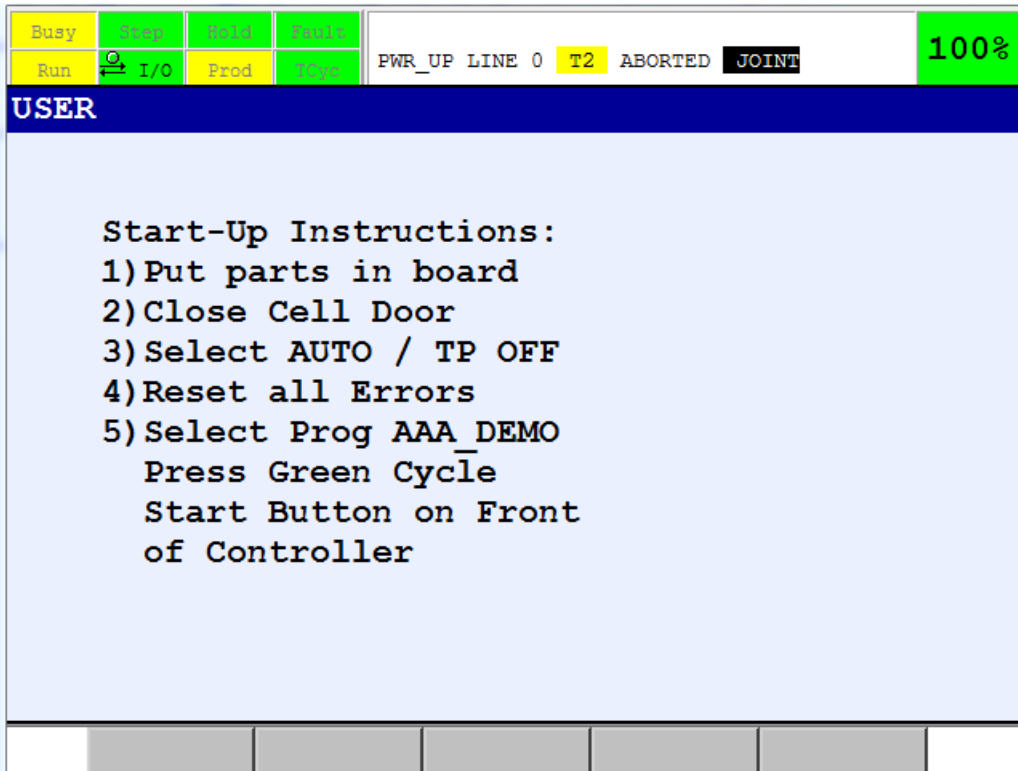
**If robot is programmed to move aggressively then this may result in
undesirable shaking of the frame,
even resulting in the activation of the door safety switch.**

Please avoid such unintended use.

1 Selecting and running Example program

1.1 Start Up Screen

When the robot controller is turned on, it displays the following Start-Up Instructions:



1.2 Example Program “AAA_DEMO”

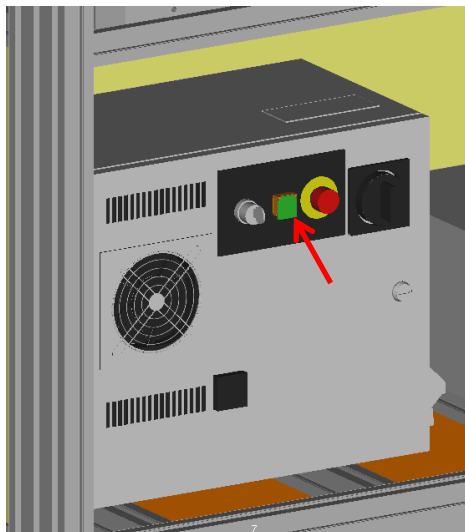
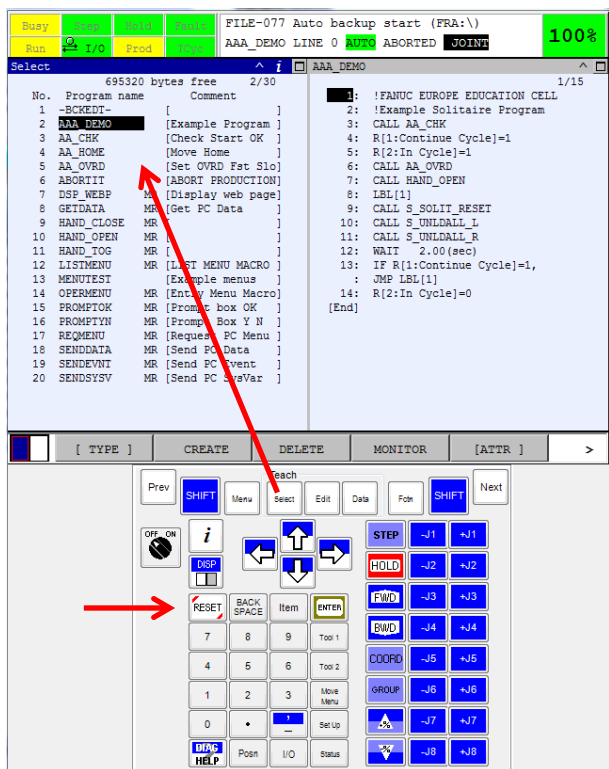
The program “AAA_DEMO” has been provided as an example program.

It transfers the cylindrical parts supplied from one triangular ‘solitaire’ board to another.

To start the program follow these steps:

- Robot must be in ‘AUTO’ mode, all E-Stops released, and cell door (‘Fence’) closed
- Robot should not be in error condition. If robot shows error condition press the “RESET” key on the Teach pendant to try to reset the errors.
- Select the program “AAA_DEMO” and Press the ‘Cycle Start’ button on the front of the controller.

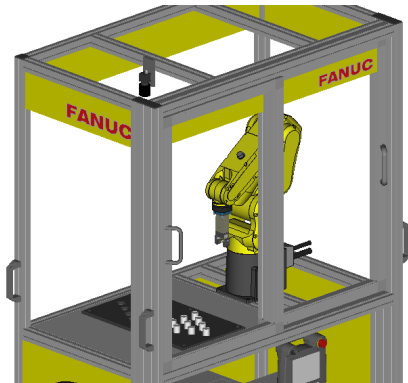
(Teach pendant enable switch must be OFF and Auto/T1/T2 switch must be in ‘Auto’ position)



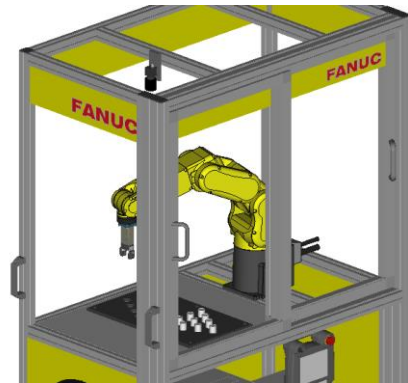
Please note – program “AAA_DEMO” must be selected before pressing ‘Cycle Start’

d. The program AAA_DEMO expects to be started from the HOME position:

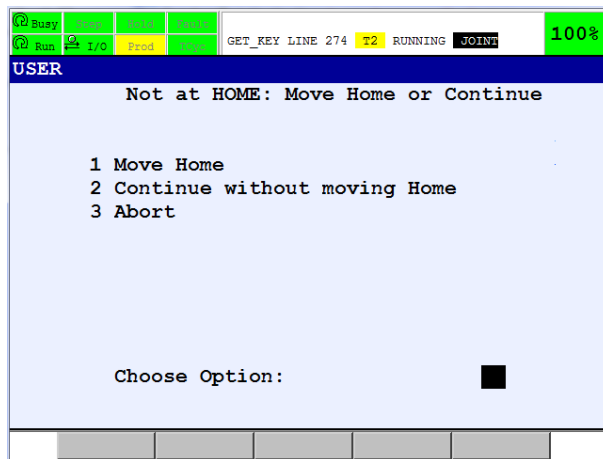
Robot at HOME



Robot not at HOME

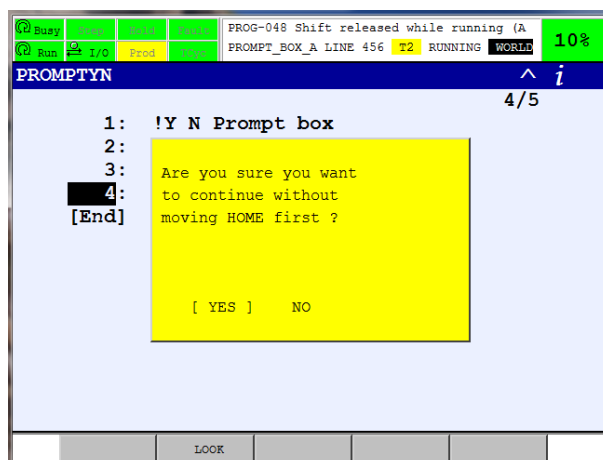


If the Robot is not at the HOME position, the following screen will appear:



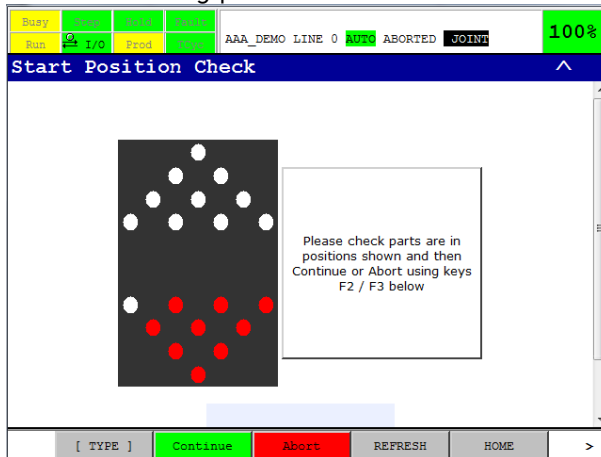
Choose the appropriate action and press 'ENTER'

If Option 2 "Continue" is chosen, then a confirmation screen will be displayed.:



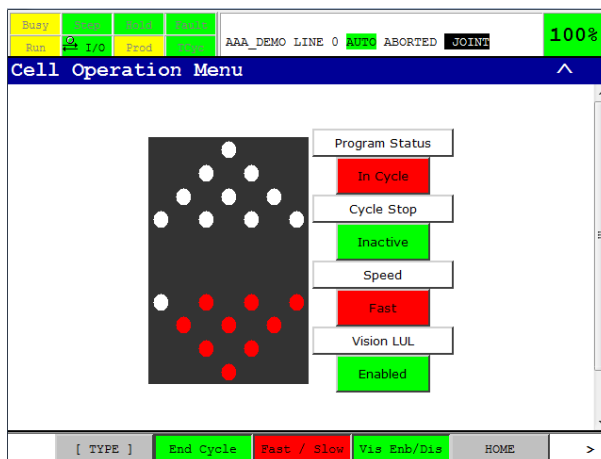
Note that these two functions have been implemented using the 'Menu Utility' function – see later section for details.

- e. Before the program starts, a screen will ask for confirmation that the parts are in the correct starting position:



Make sure that the parts are in the positions shown, then press "Continue".

- f. When the program is running, a Status / Menu screen is shown:.

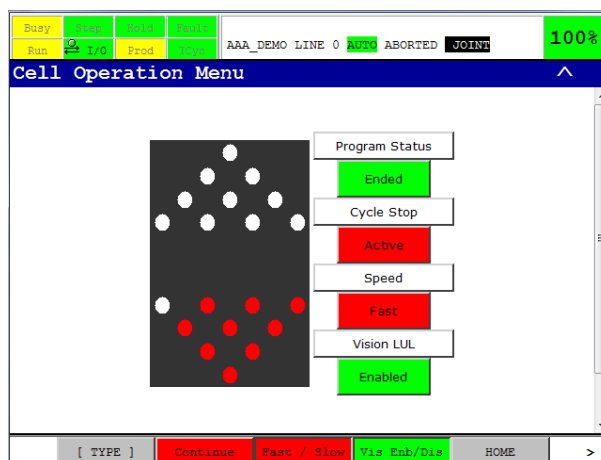


The red/white circles show the current position of the pieces

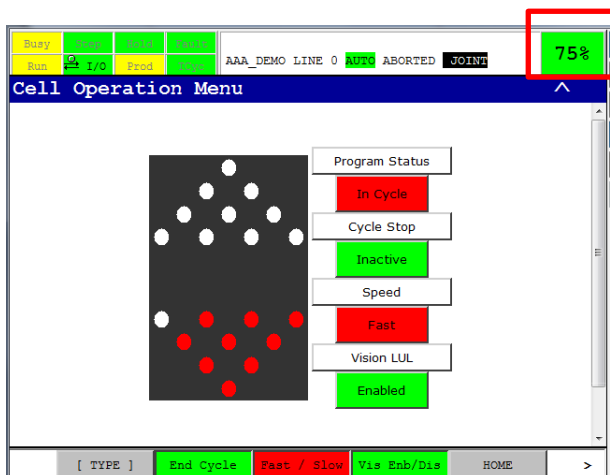
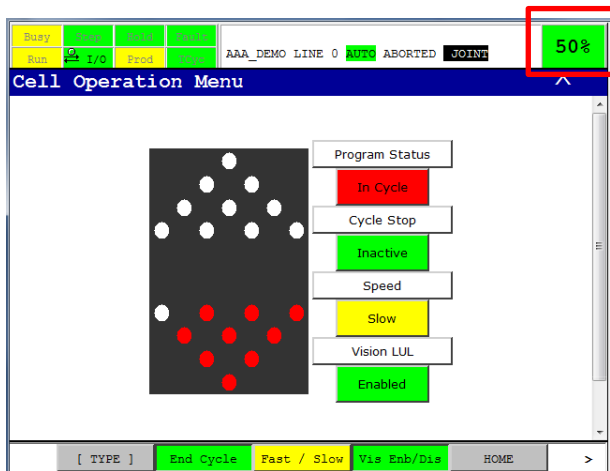
There are two function Key Selections: End Cycle / Continue and Fast/Slow.

If 'End Cycle' is pressed, this changes Register R[1] from 1 to 0, and the Cycle Stop status changes from Inactive to Active:

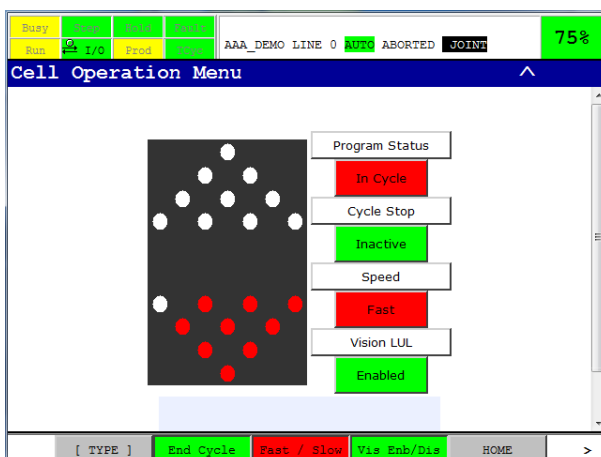
This will mean that the program will stop at the end of the next complete cycle and the Program Status will then become 'Ended'



g. Likewise, the Fast/Slow Function key changes the Override:



Note: the Fast / Slow Override speeds are set according to the values in R[4] and R[5] – see later section for details

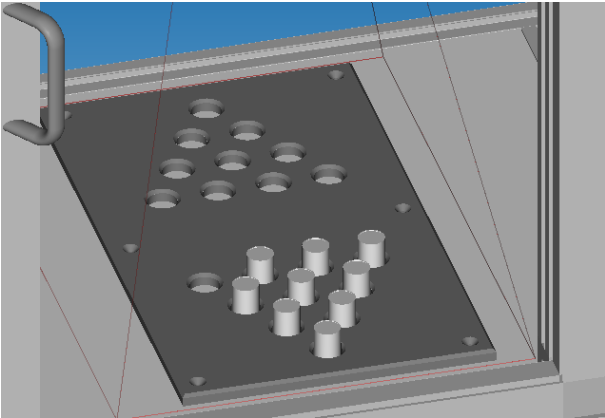


And the F4 function Key enables / disables the Vision Load / Unload section of the program – see next section for details.

Note: *iR*Vision must be installed and set up for this function to be used.

1.3 AAA_DEMO Program Sequence with / without iRVision

There are 3 basic positions for the parts in/on the Solitaire board:



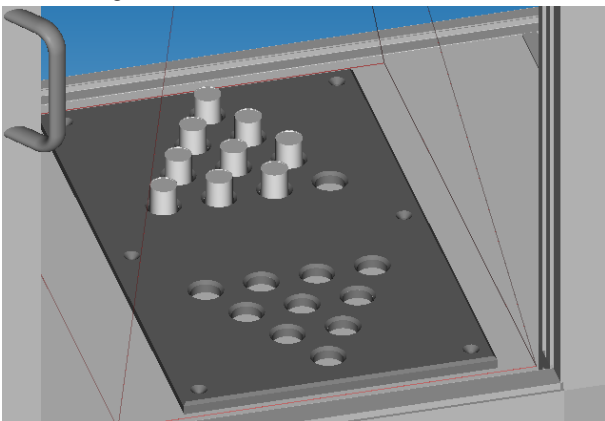
In the Left Hand side of the board:

This is the start and end position of the example program

If Vision L/UL is Disabled then sequence is:

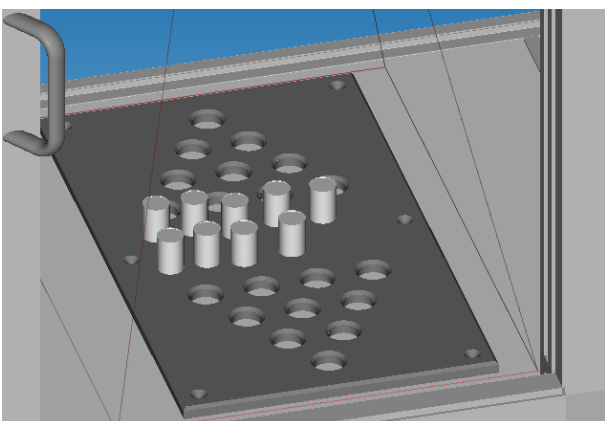
- Move Parts from Left Hand side to Right Hand side
- Move Parts from Right Hand side to Left Hand side
- Repeat

In the Right Hand side of the board:



If Vision L/UL is Enabled then sequence is:

- Move Parts from Left Hand side to Right Hand side
- Move Parts into middle of board
- Move Parts from middle of board to Left Hand side using iRVision.
- Repeat



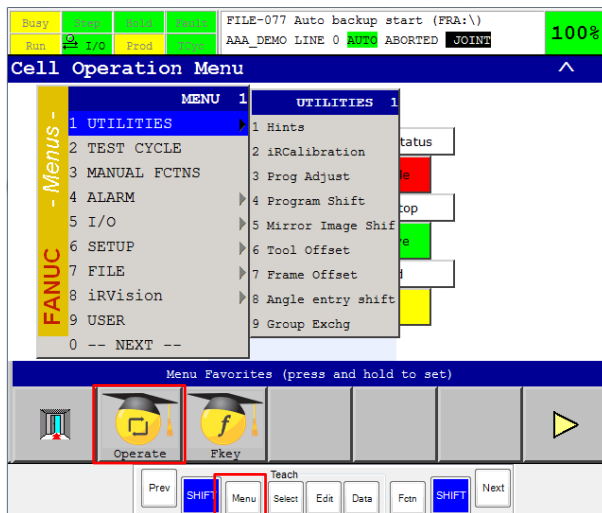
On

the middle of the board:

This position is only used if iRVision is enabled for the example program using "F4" function key

1.4 Operation Menu Display

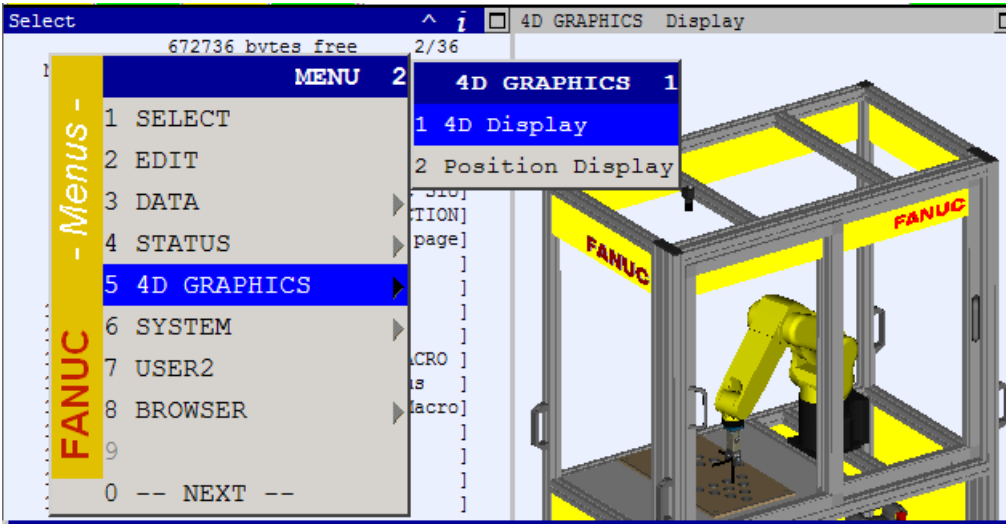
The Operation Menu can be displayed at any time by pressing the MENU key and then selecting the “Operate” shortcut that pops up:



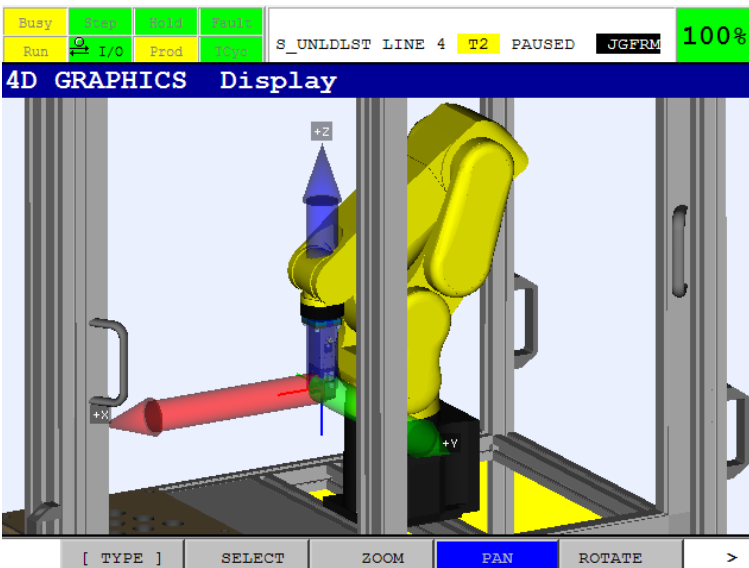
1.5 4D Graphics (Option)

The “4D Graphics” function is available as an option, with a model of the cell loaded into the robot controller.

This model can be displayed by pressing the MENU button and selecting “4D Graphics”:



The 4D graphics function has many features, for example as shown below the display of the Jog Coordinates – in this case the WORLD Jog Coordinate System:



2 Demonstration Program Setup / Teaching

The positions of the holes in the tray are calculated, not taught – so there should be no need to touchup / reteach the individual hole positions. However, it may be necessary to touchup / reteach the User Frame (UFrame) which defines the position of the tray relative to the robot.

2.1 Gripper Setup

To teach the positions it is necessary to use a part held in the gripper.

Gripper Force should be set at 50% using the small rotary switch on the gripper.

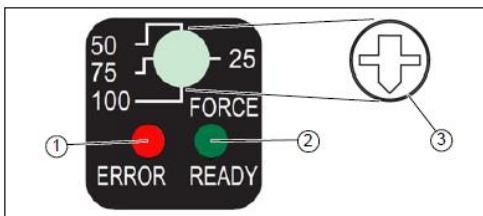


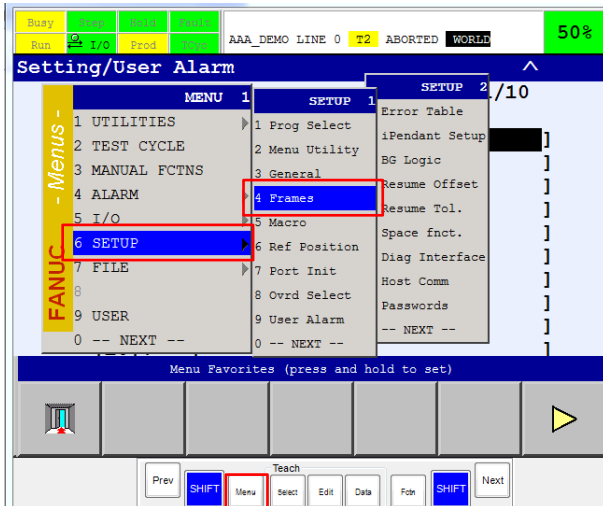
Fig. 4 Function label with rotary switch of EGP 40

1	Error-LED
2	Power-On-LED
3	Rotary switch (not available for EGP-S)

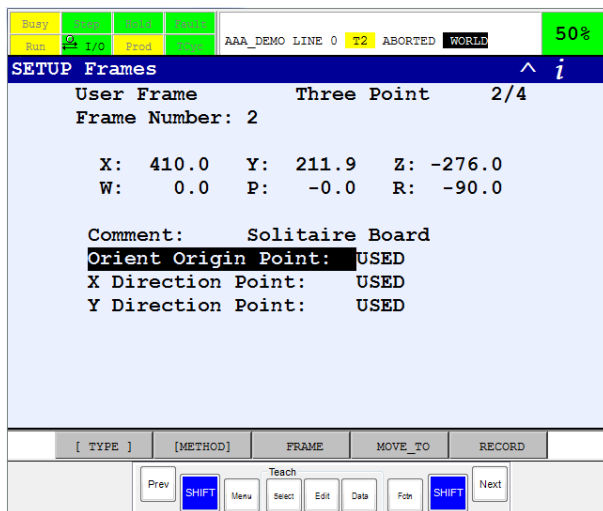


2.2 UFrame 2 Setup

The User Frame touchup is done using the Menu>Setup>Frames function:



This requires 3 Points to be touched-up / re-taught:

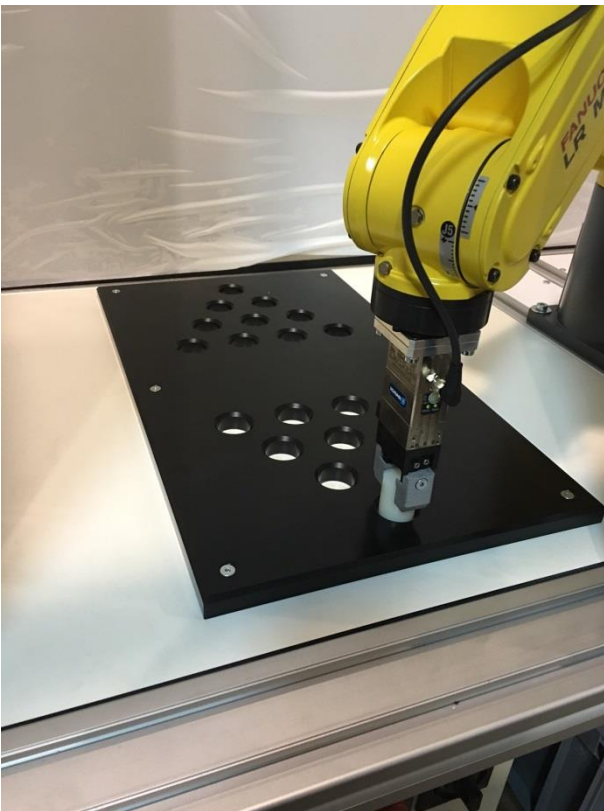


Note that the MOVE_TO function key provides a method to easily check the current taught positions

First, insert a piece into the gripper:

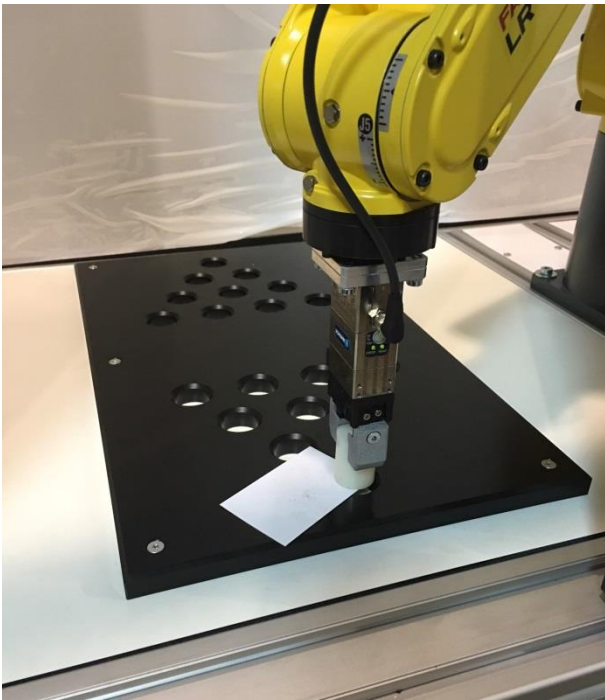


Make sure gripper is vertical, and jog the robot so that the piece is central in the Left-most hole on the tray shown below:



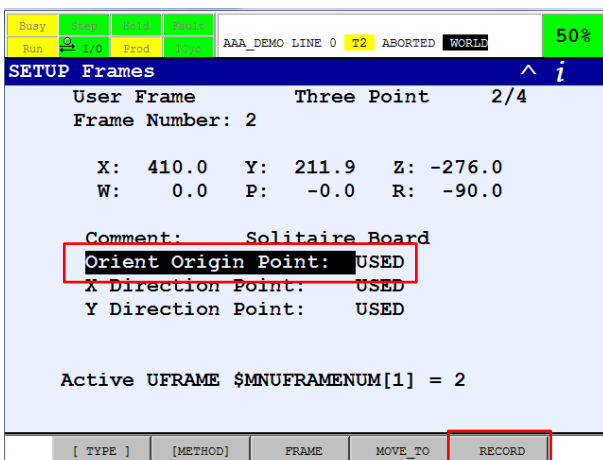
It is difficult to know when the piece is touching the table at the bottom of the hole – so this position is not used for the reference point.

Instead, jog the robot upwards using WORLD+Z until the bottom of the piece is just above the top of the tray. Use a thin piece of card to help judge this correctly:



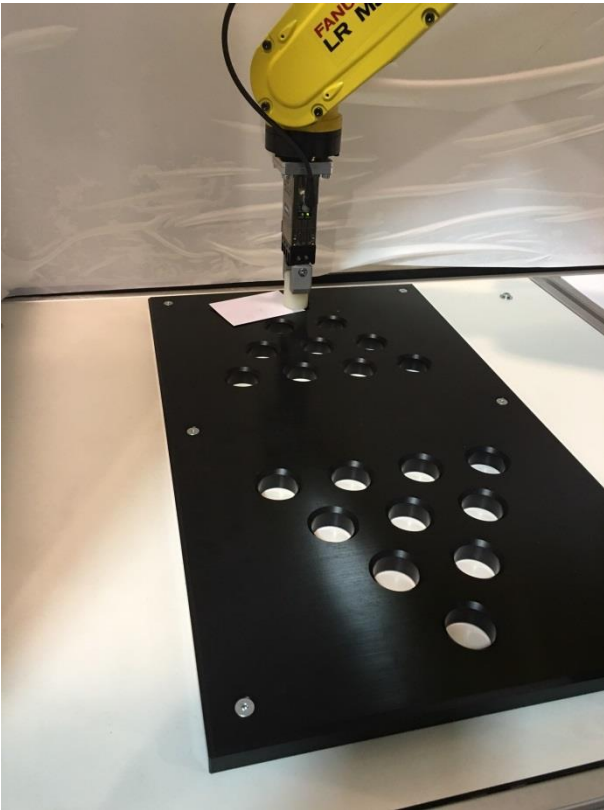
2.3 Orient Origin Point

Then RECORD this position as Orient Origin Point – see below:

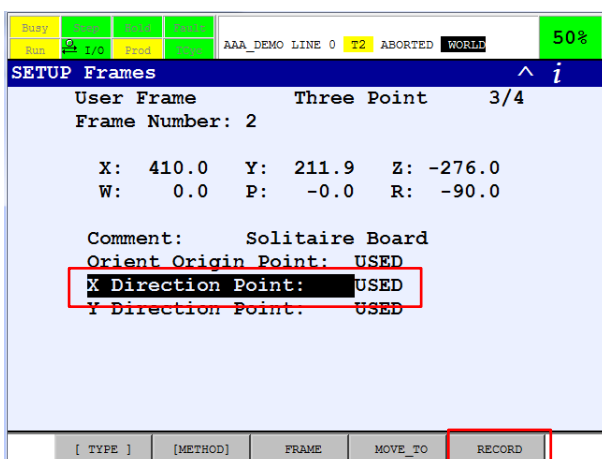


2.4 X-Direction Point

Now do the same thing for the Right-most hole in the tray:



And RECORD this position as X-Direction Point – see below:

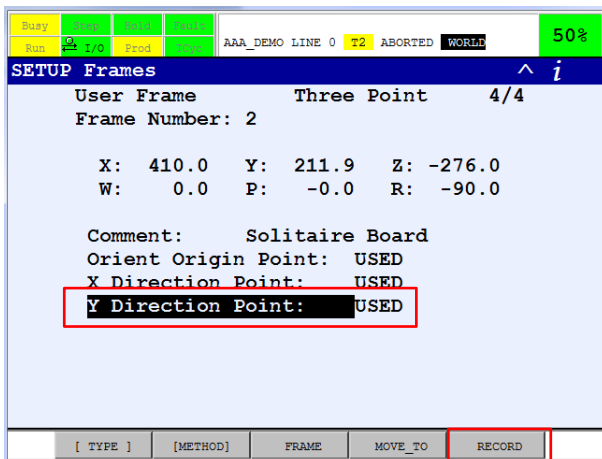


2.5 Y-Direction Point

And finally any position on the top surface of the front of the tray:



And RECORD this position as Y-Direction Point – see below:



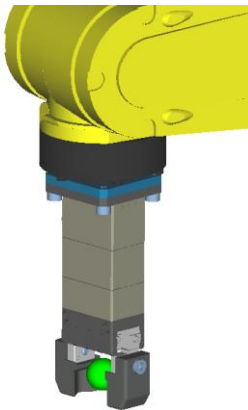
Note – Y-Direction Point defines the X-Y plane, so any position in the right plane will do.

This procedure will touch-up / re-teach the UFrame #2

3 Robot Setup

3.1 UTOOL / TCP Setup

Because of the simple shape and mounting of the Schunk gripper, a simple TCP with an offset of 115mm in Z is sufficient.



Tool Frame	Direct Entry	1/7
Frame Number: 1		
1	Comment:	EGP40
2	X:	0.000
3	Y:	0.000
4	Z:	115.000
5	W:	0.000
6	P:	0.000
7	R:	0.000
Configuration:		N D B, 0, 0, 0
Active TOOL \$MNUTOOLNUM[1] = 1		

3.2 Payload Setup

The payload is quite low for this application – so only one payload of 0.5 kg has been set:

Group	Number	Default	Help
Group 1			
1	Schedule No[1]:	EGP40	
2	PAYLOAD	[kg] 0.50	
3	PAYLOAD CENTER X	[cm] 0.00	
4	PAYLOAD CENTER Y	[cm] 0.00	
5	PAYLOAD CENTER Z	[cm] 5.00	
6	PAYLOAD INERTIA X	[kgfcm ²] 0.00	
7	PAYLOAD INERTIA Y	[kgfcm ²] 0.00	
8	PAYLOAD INERTIA Z	[kgfcm ²] 0.00	

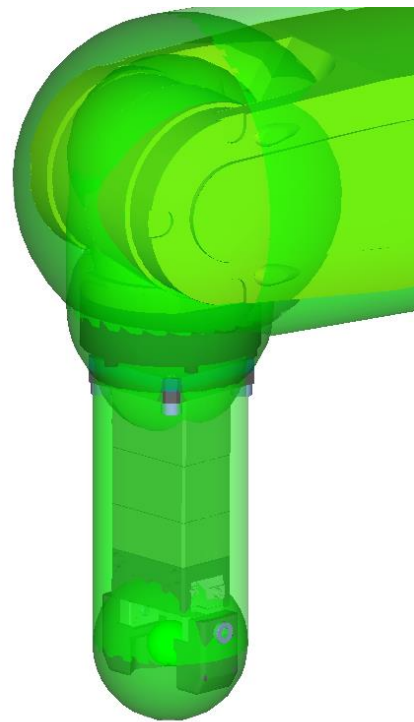
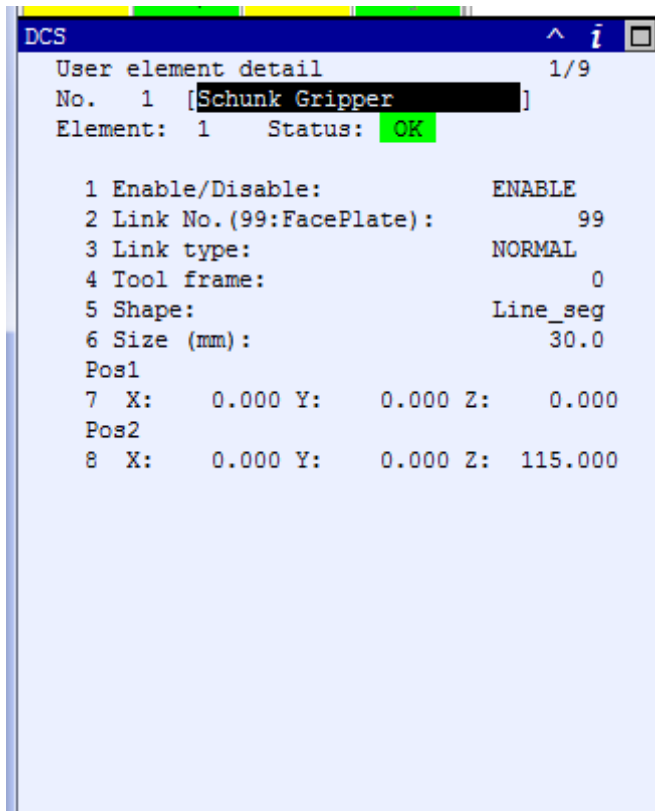
3.3 DCS Setup

DCS (Dual Check Safety) has been used to ensure that the robot cannot accidentally hit the walls of the cell. This requires setting up a Tool Model and a Safe Zone.

3.3.1 DCS Tool Model.

A simple DCS model using one "Line_seg" type model has been used.

For more details please refer to DCS manuals.

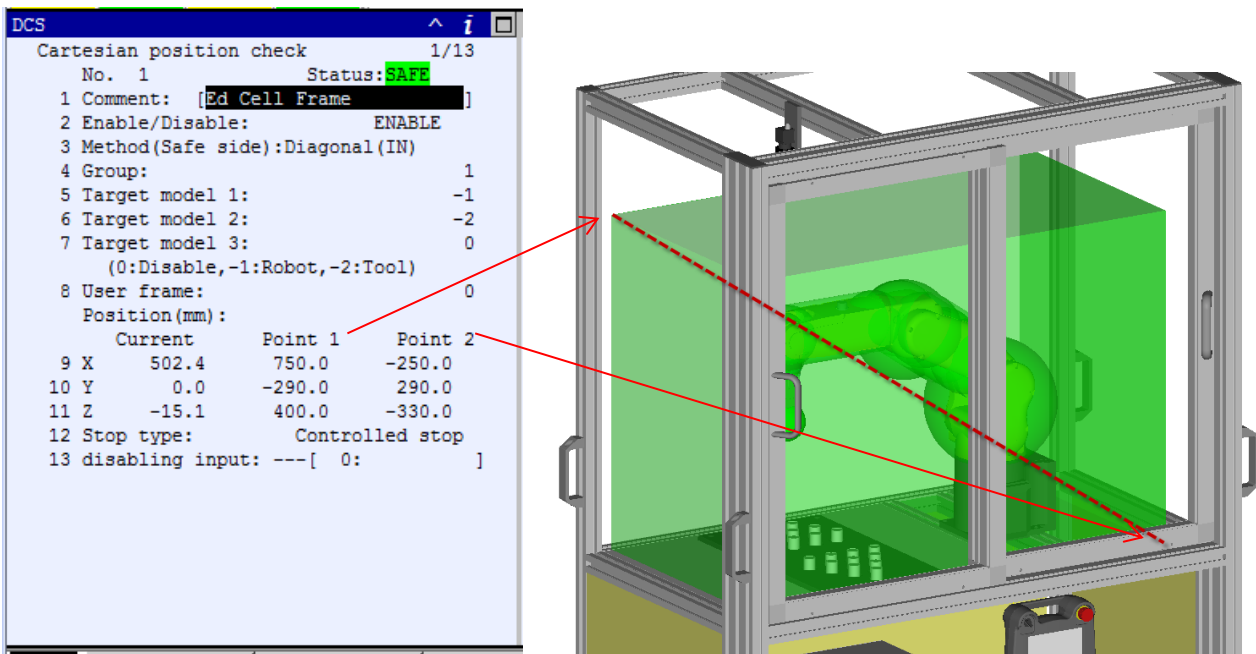


3.3.2 DCS Safe Zone

A simple DCS Cartesian Position Check Zone has been set up using a Diagonal line to define a cuboid zone, running from a point on the “ top left “ of the cell to the “bottom right “ – see screenshot below - where the inside of the cuboid is safe.

If the robot or the tool comes close to the edge of this zone, the robot will stop.

For more details please refer to DCS manuals.



3.4 I/O Setup

Only 2 I/Os are used, for gripper Open / Close

These outputs are configured as Complementary, so when, for example RO[7] is set ON, RO[8] is automatically forced to OFF and vice-versa.

#	SIM	STATUS	
RO[1]	U	OFF	[]
RO[2]	U	OFF	[]
RO[3]	U	OFF	[]
RO[4]	U	OFF	[]
RO[5]	U	OFF	[]
RO[6]	U	OFF	[]
RO[7]	U	ON	[Open Gripper]
RO[8]	U	OFF	[Close Gripper]

Port Detail 1/3

Robot Dig. Output [7]

1 Comment: [Open Gripper]

2 Polarity: NORMAL

3 Complementary: TRUE [7 - 8]

[TYPE] IN/OUT ON OFF >

3.5 Macro Setup

One macro has been set up, to allow the user to easily toggle the gripper open / close using SHIFT + User Key 1 on the Teach Pendant:

The screenshot displays the FANUC CNC control interface. At the top, the status bar shows 'HAND_TOG LINE 0 AUTO ABORTED JOINT' and '100%'. Below this, the 'Macro Command' window is open, showing the macro 'HAND_TOG' with 1/150 instructions. The first instruction is highlighted with a red box: '1 [Open Close Hnd] [HAND_TOG]SU[1]'. The macro code is as follows:

```

1: !Toggle Hand Open/Close
2: IF RO[7:Open Gripper]=ON,
: JMP LBL[1]
3: RO[7:Open Gripper]=ON
4: JMP LBL[2]
5: LBL[1]
6: RO[7:Open Gripper]=OFF
7: LBL[2]
[End]

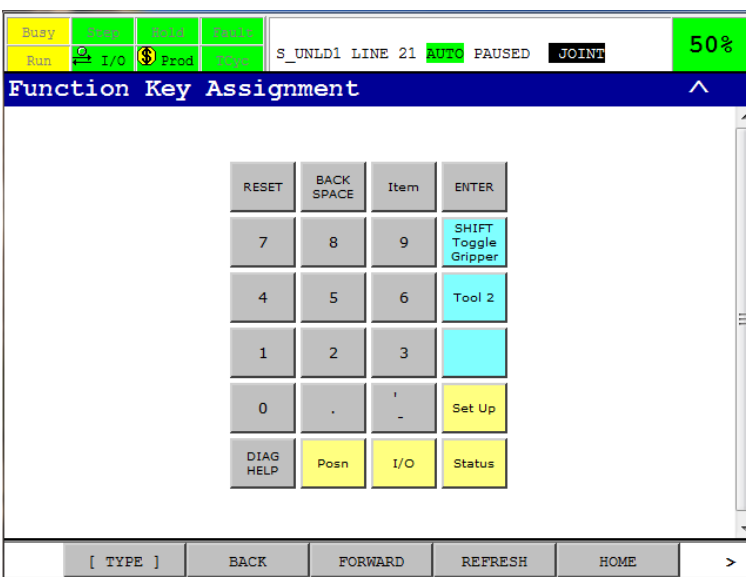
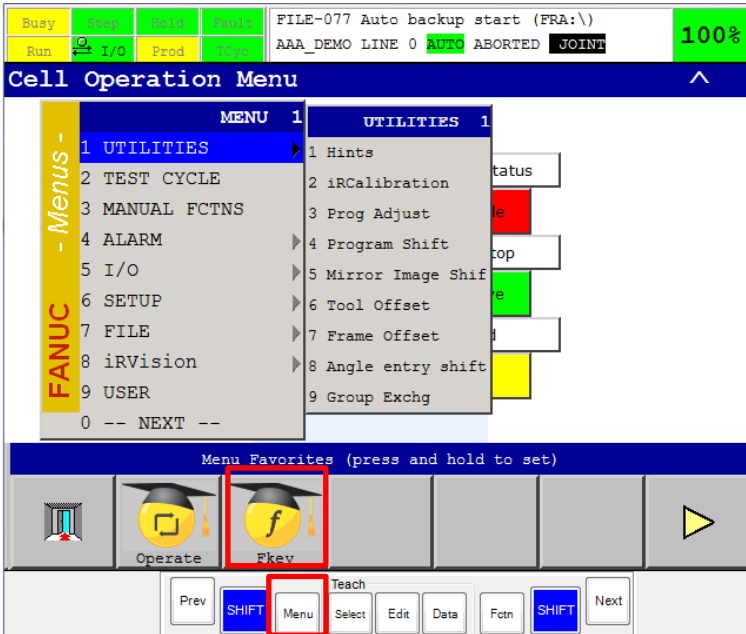
```

Below the macro editor, the 'Teach' panel is visible. The 'SHIFT' button is highlighted with a red box. The 'ENTER' button is also highlighted with a red box. The 'HOLD' button is highlighted with a red box. The 'Tool 1' button is highlighted with a red box.

3.6 TP Function Key Hint Screen

To allow the user to easily remember the setting of the Function Key, a Hint Screen has been set up using an HTML Page:

When the “Menu” key is pressed, the Shortcut key F2 “FKey” pops up:



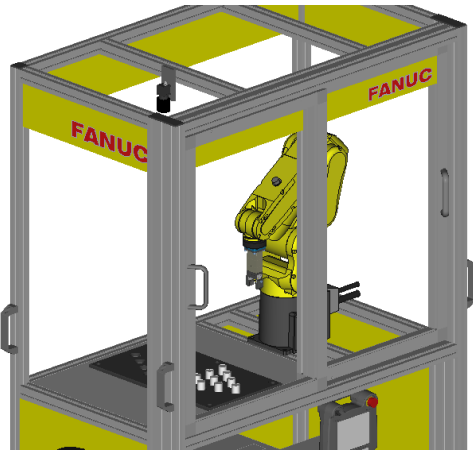
Pressing F2 will display the FKey Hint screen:

This screen shows the current assignment of the function keys.

The text which is displayed is the text stored in String Registers SR[21] – SR[25]

3.7 Home / Reference Position

One reference position has been set up, which is used for the HOME position check. Home position is this:



And setup is this:

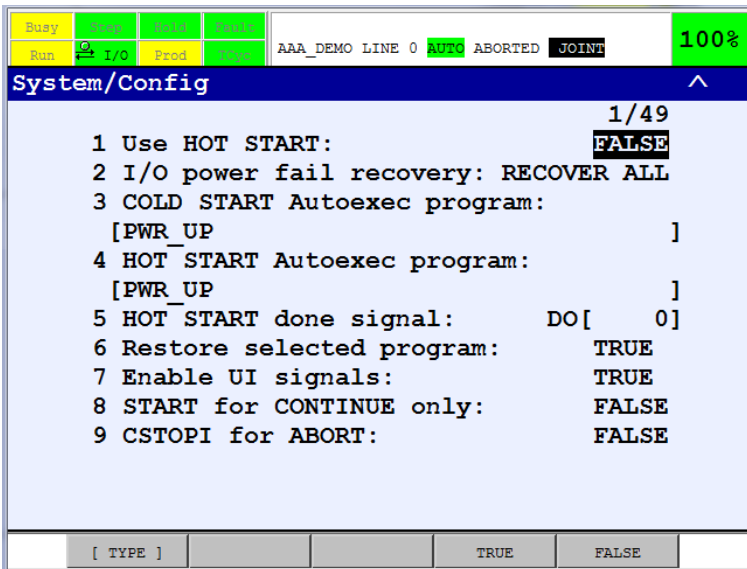
Busy	Ready	Auto	Manual	100%
Busy	Ready	Auto	Manual	100%
Run	I/O	Prod		
AAA_DEMO LINE 0 AUTO ABORTED JOINT				
REF POSN 1/13				
Reference Position				
Ref.Position Number: 1				
1 Comment: [Home Position]				
2 Enable/Disable: ENABLE				
3 Is a valid HOME: TRUE				
4 Signal definition: DO [101]				
5	J1:	-0.000	+/-	2.000
6	J2:	-28.000	+/-	2.000
7	J3:	-35.000	+/-	2.000
8	J4:	0.000	+/-	2.000
9	J5:	-55.000	+/-	2.000
10	J6:	0.001	+/-	2.000
[TYPE]				RECORD

Busy	Ready	Auto	Manual	100%
Busy	Ready	Auto	Manual	100%
Run	I/O	Prod		
AAA_DEMO LINE 0 AUTO ABORTED JOINT				
I/O Digital Out 101/512				
#	SIM	STATUS		
DO[96]	*	*	[]
DO[97]	*	*	[]
DO[98]	*	*	[]
DO[99]	*	*	[]
DO[100]	*	*	[]
DO[101]	U	ON	[HOME Signal]
DO[102]	U	OFF	[]
DO[103]	U	OFF	[]
DO[104]	U	OFF	[]
DO[105]	U	OFF	[]
DO[106]	U	OFF	[]
[TYPE]	CONFIG	IN/OUT	ON	OFF >

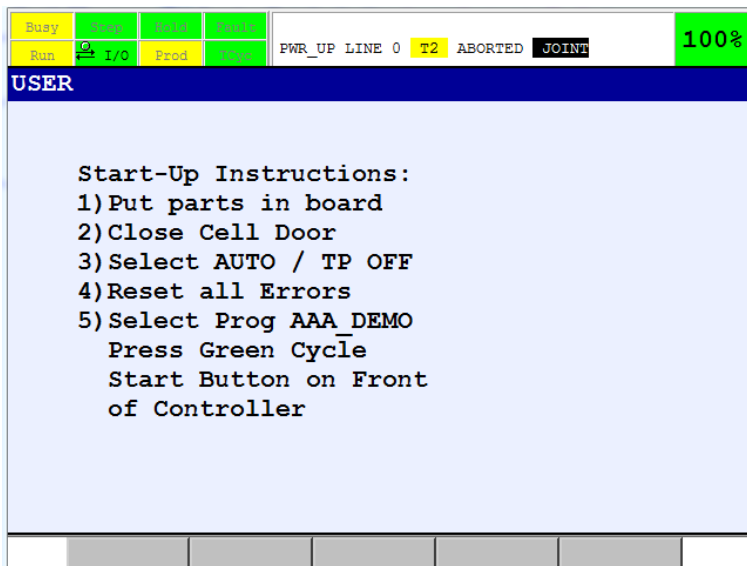
When the robot is in this position, the output DO [101] will be ON – and this can be checked by the TP program AA_CHK

3.8 Power Up Program

To provide instructions to the user, a Power-Up program “PWR_UP.TP” has been assigned to both Hot Start and Cold Start.



The program simply writes instructions to the Teach Pendant:



4 iRVision Setup (Option)

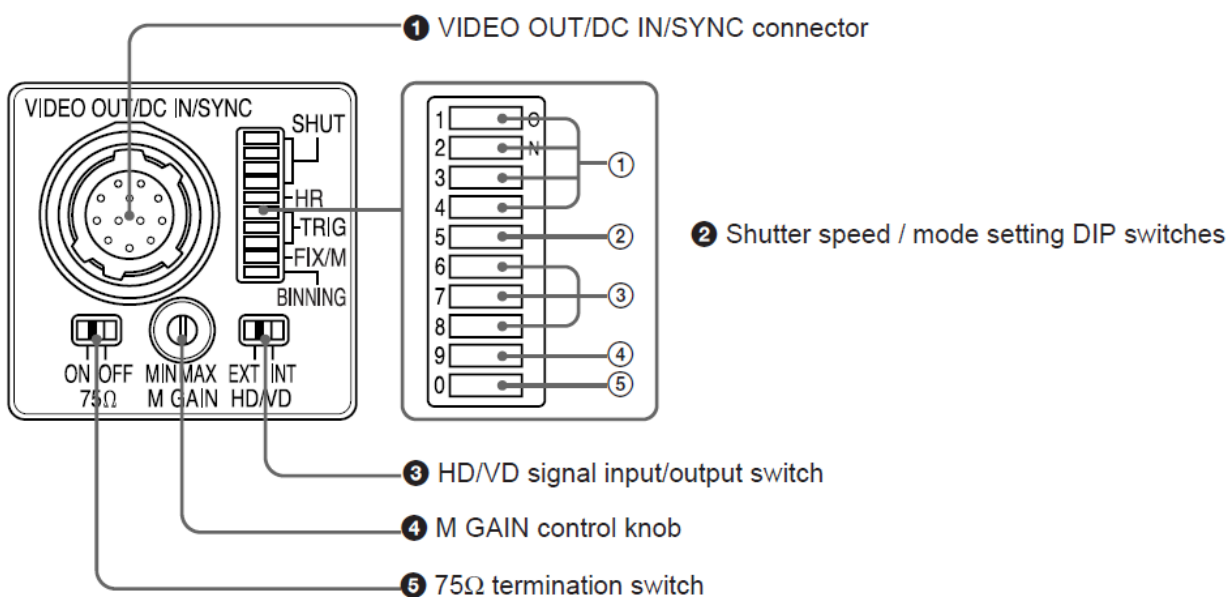
The Education Cell is available with and without integrated iRVision using Sony XC56 video camera.

In either case, the iRVision Setup has not been done on the Education Cell controller. Please contact your local FANUC representative to arrange training on iRVision

The following section is just to give a short overview of a sample iRVision setup, and is not intended to take the place of a proper iRVision training.

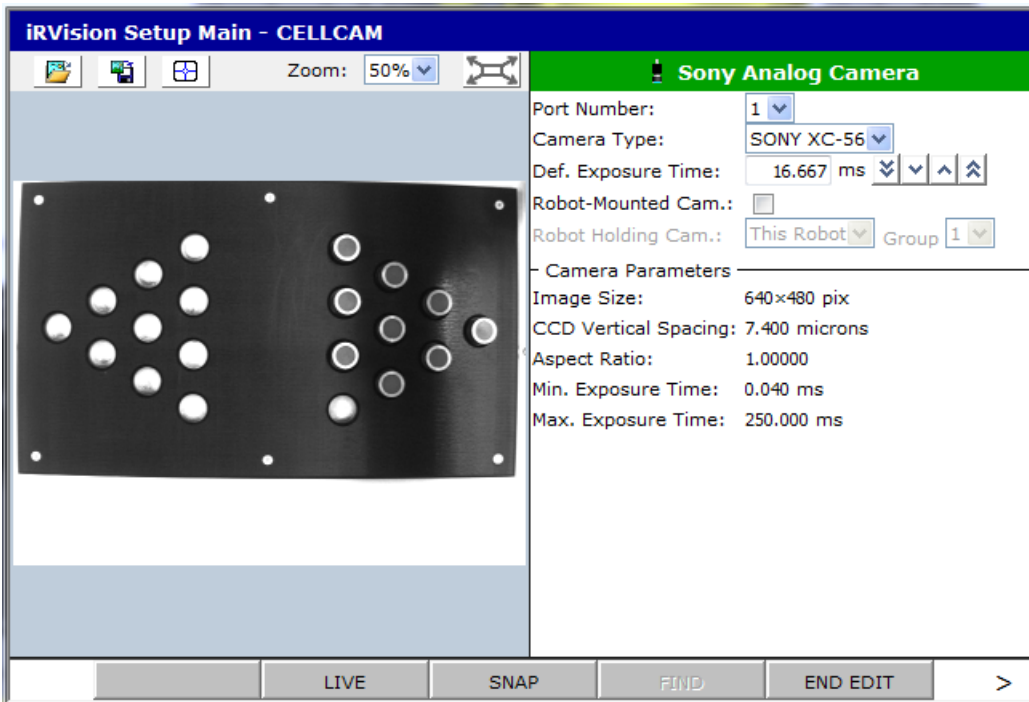
4.1 Camera Adjustment

In order to work correctly, the DIP switches on the XC-56 Camera should be set as shown:

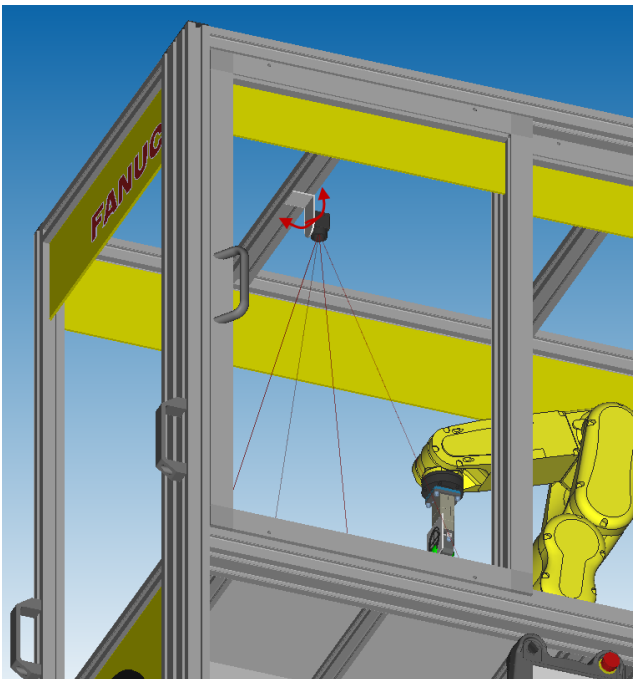


Dip Switch	Setting	Dip Switch	Setting
1	OFF	EXT trigger	ON
2	OFF	75 Ohm Switch	ON
3	OFF		
4	OFF		
5	OFF		
6	OFF		
7	ON		
8	ON		
9	OFF		
10	OFF		

If this option has been ordered, the basic camera setup should already have been done by FANUC Europe, so the camera should display an image something like this:



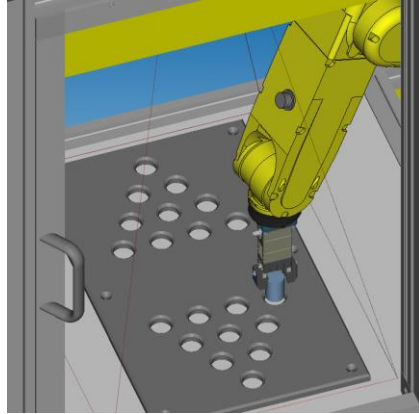
If the Camera View is not like this, then please adjust using the screws on the Camera Mounting bracket:



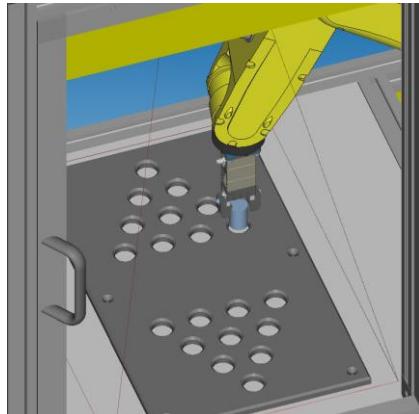
4.2 Application Frame

It is good practice to set up an Application Frame to use with *iR*Vision, for example Uframe 9. This can be done in the same way as shown in section 2:

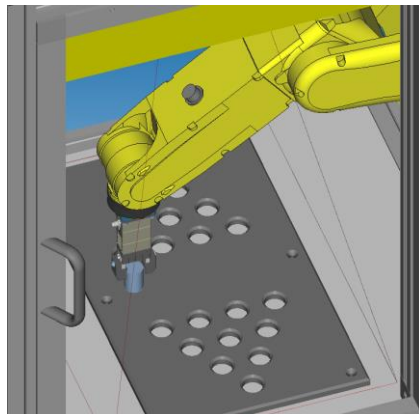
Orient Origin Point:



X-Direction Point:



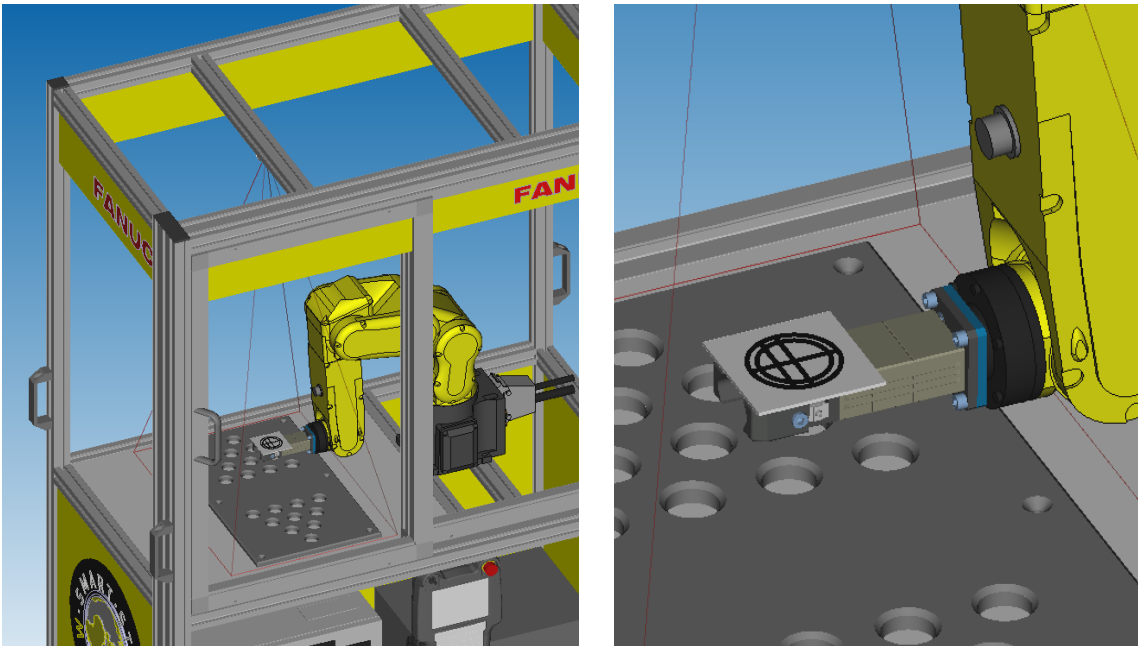
Y- Direction Point:



4.3 Camera Calibration

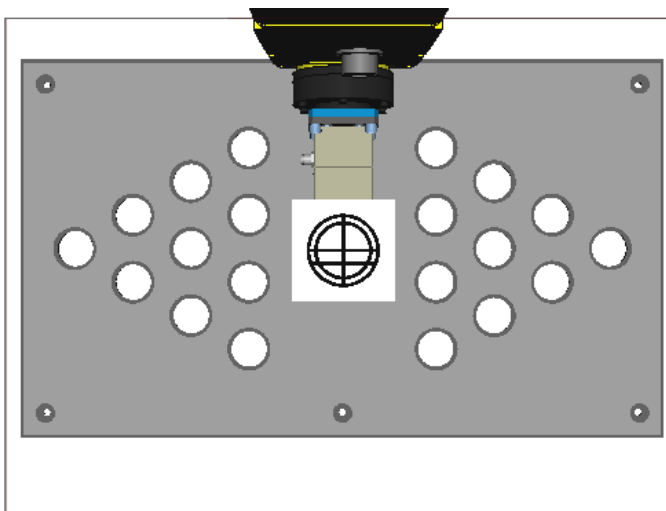
The *iR*Vision Calibration Grids are not included in the Education Cell. Instead the “Robot Generated Grid Calibration” Method can be used – please refer to the *iR*Vision Operation Manual for details.

Basically a target should be temporarily attached to the Gripper as shown:



[The design of the target is explained in the *iR*Vision manual – it should be approx. 50mm diameter]

It will appear something like this in the camera field-of-view:

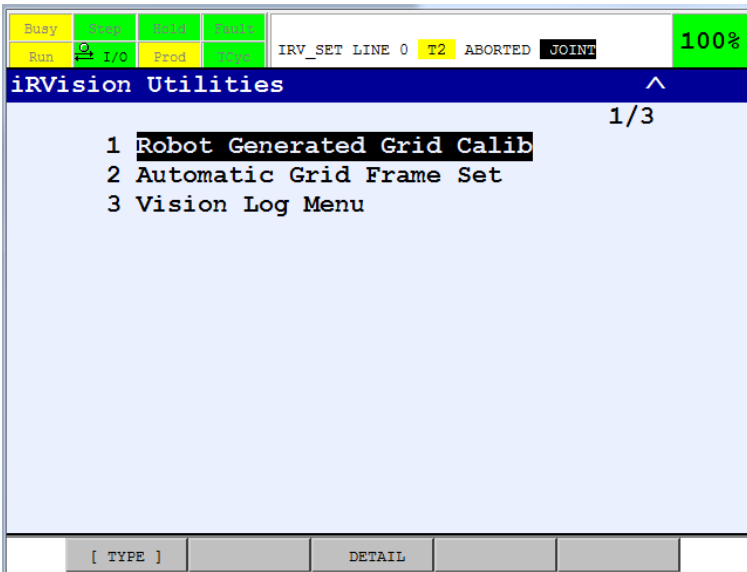


The Camera Calibration takes place in 2 steps:

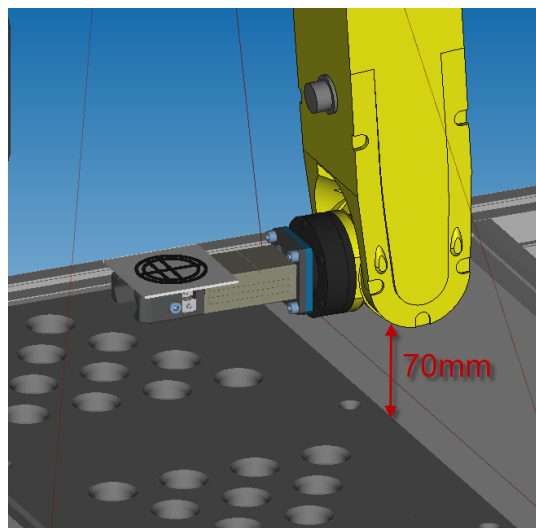
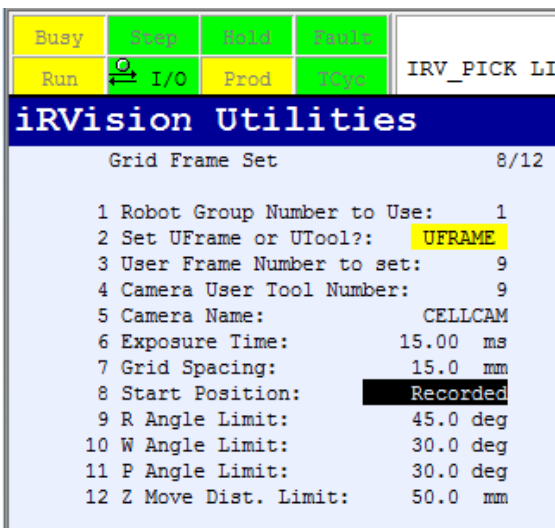
- First find the relationship of the target to the robot
- Second move the target around the field of view to calibrate the camera

Please note that it may be necessary to disable DCS while calibrating the camera field of view since the robot moves the target right to the edges of the field of view, close to the side walls.

All of this is done through the Teach Pendant using the *iR*Vision Utilities Menu:



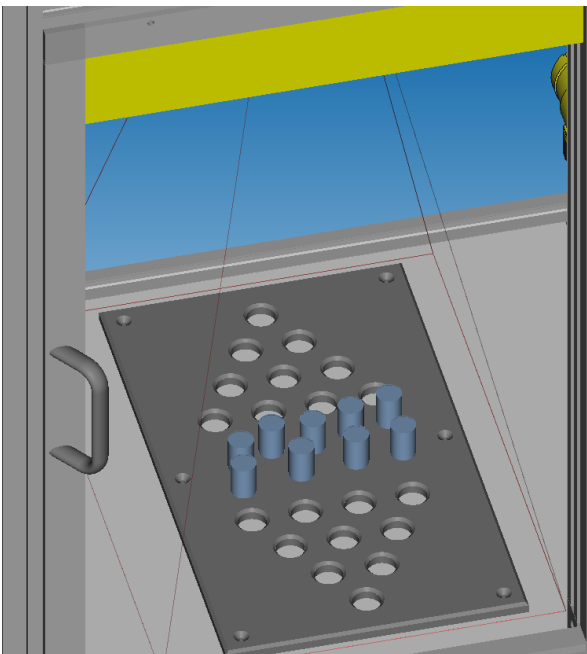
For details please refer to the *iR*Vision Operation Manual. Note that for the Start Position, there must be enough space between the robot wrist and the cell – approx. 70mm is good:



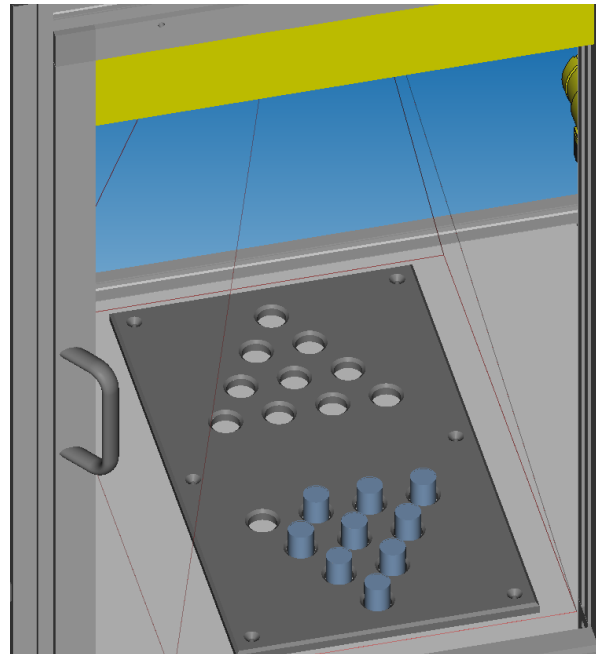
4.4 Example *iR*Vision Application

A simple application could be to load the parts from the centre of the board into the starting position for the Solitaire program

So the parts start like this,
randomly placed in the centre of the board



And end like this,
in the start position for 'Solitaire'



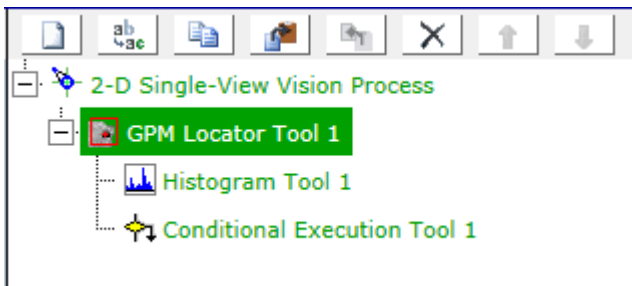
Using FANUC *iR*Vision this is quite straightforward to do:

- Calibrate the Camera (as shown previously)
- Teach a reference pick-up position for one of the parts
- Teach *iR*Vision to recognise and locate one of the parts.
- Create a TP program to use the *iR*Vision information to pick the part and then place into the board.

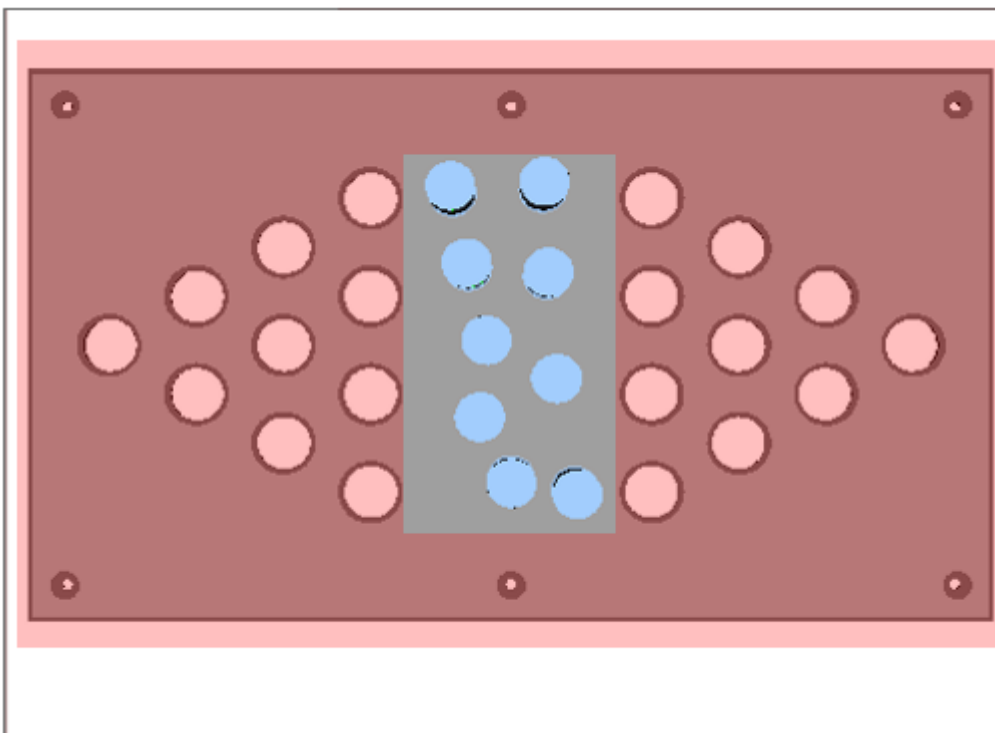
(This was suggestion but has been now added to the demo cell as shipped by FANUC – see section 1.3)

4.5 iRVision Tips

FEC set this application up using the Vision Process below:

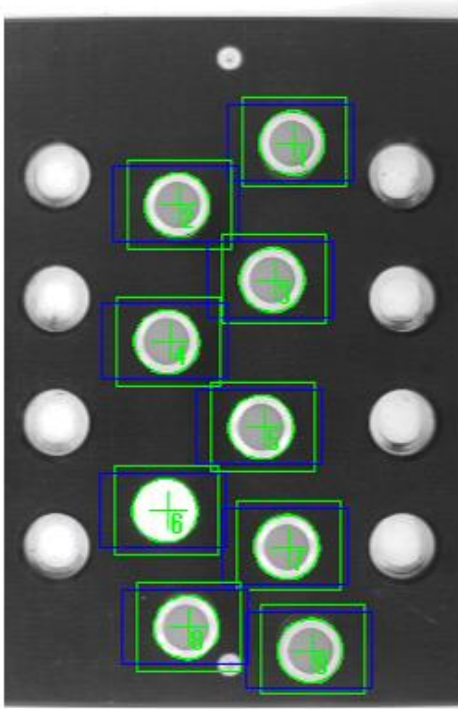


- A simple GPM locator tool to find the cylindrical part
- The GPM Locator has a run-time mask to restrict the vision to the centre of the board to avoid finding the 'holes' instead of the parts:

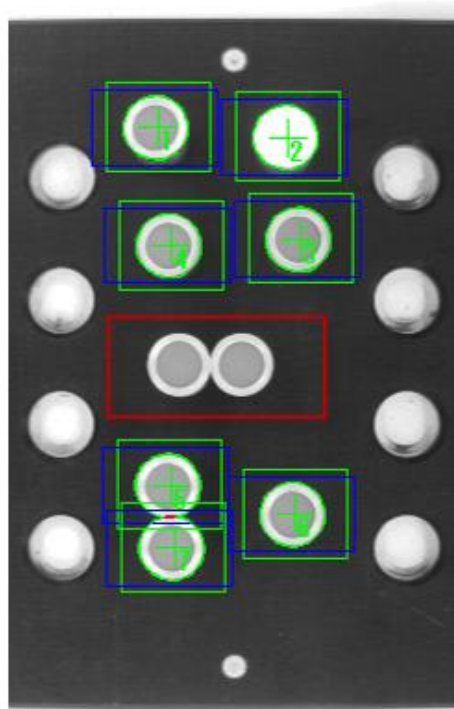


- Then the Histogram Tool and the Conditional Execution Tool are used to make sure that there is enough free space around the part to allow it to be picked up.

So all of these parts can be picked:



But the two central parts here cannot be picked because they are too close together.



4.6 TP Program

Write a simple TP program to Pick and Place the parts:

```

1: !Pick Part with Vision
2:
3: UTOOL_NUM=1
4: UFRAME_NUM=9
5:
6: LBL[1]
7:
8: VISION RUN_FIND 'FIND_CYL'
9: VISION GET_OFFSET 'FIND_CYL' VR[1] JMP LBL[90]
10: CALL HAND_OPEN
11:J P[1] 25% FINE VOFFSET,VR[1] Tool_Offset,FR[40:ToolOffset Abo]
12:L P[1] 500mm/sec FINE VOFFSET,VR[1] Tool_Offset,FR[41:ToolOff PickP1]
13: CALL HAND_CLOSE
14:L P[1] 500mm/sec FINE VOFFSET,VR[1] Tool_Offset,FR[40:ToolOffset Abo]
15:
16: JMP LBL[99]
17: LBL[90]
18: !Ask operator to fix
19:L P[2] 500mm/sec FINE
20: CALL PROMPTOK(2)
    
```

The screenshot shows a FANUC TP program editor window titled "IRV_PICK". The program code is displayed in a list format from line 1 to 20. Line 1 is highlighted in yellow and contains the comment "!Pick Part with Vision". Line 18 is also highlighted in yellow and contains the comment "!Ask operator to fix". The status bar at the top indicates "CVIS-218 The parent tool failed to find" and "IRV_PICK LINE 0 AUTO ABORTED JOINT" with a "100%" completion indicator. The bottom of the window has buttons for "POINT" and "TOUCHUP".

It is advisable also to add in some simple error handling to check that the Vision is working OK. This can be done using the Menu Utility as described in section 7

```

Prompt msg menu no: 2 1/6
Menu name: VisFail
Line Text
1 Vision failed to find Part
2 Check Parts placed OK
3
4 Press OK to Retry
5
[ OK ]
    
```

The screenshot shows the "SETUP Menu Utility" window. It displays a menu configuration for a custom error handling routine named "VisFail". The menu has 5 lines: Line 1 is "Vision failed to find Part", Line 2 is "Check Parts placed OK", Line 3 is blank, Line 4 is "Press OK to Retry", and Line 5 is blank. A yellow dialog box is overlaid on the screen, showing the text "Vision failed to find Part", "Check Parts placed OK", "Press OK to Retry", and "[OK]". The status bar at the top is the same as in the previous screenshot. The bottom of the window has buttons for "[TYPE]", "TEST", and "HELP".

For more details please refer to the *iR*Vision Operation Manual or arrange training with your local FANUC Europe representative.

5 List of Registers

5.1 Numeric registers

Registers are used to store settings to control the cell operation, and used by the program internally.

Register	Comment	Description	Default value
1	Continue Cycle	This register is set to 1 at start of "AAA_DEMO". If value is 1 then the program will loop continuously. If value is 0, program will stop at end of next full cycle. Value can be set manually or by the Menu screen described in earlier section.	1 to loop 0 to end
2	In Cycle	This register is set to 1 at start of "AAA_DEMO", and is set to 0 at end of program	1 when in cycle 0 at end of cycle
3	Speed Mode	This register is set via the Menu screen described earlier.	1 is fast mode, anything else slow.
4	Fast OVRD	This is value that will be used for Override in Fast Mode	75%
5	Slow OVRD	This is value that will be used for Override in Slow Mode	50%
6	User Input	Used by the Menu Utility to return the User's Choice of actions	1
7	Vision Installed	This should be set during software installation / setup.	1 is installed, anything else not installed
8	Vision Enabled	This is set by the Operation Menu to enable / disable Vision L/UL. Both R[7] and R[8] must be 1 for the vision section of the example program to run	1 is enabled, anything else not enabled

FANUC

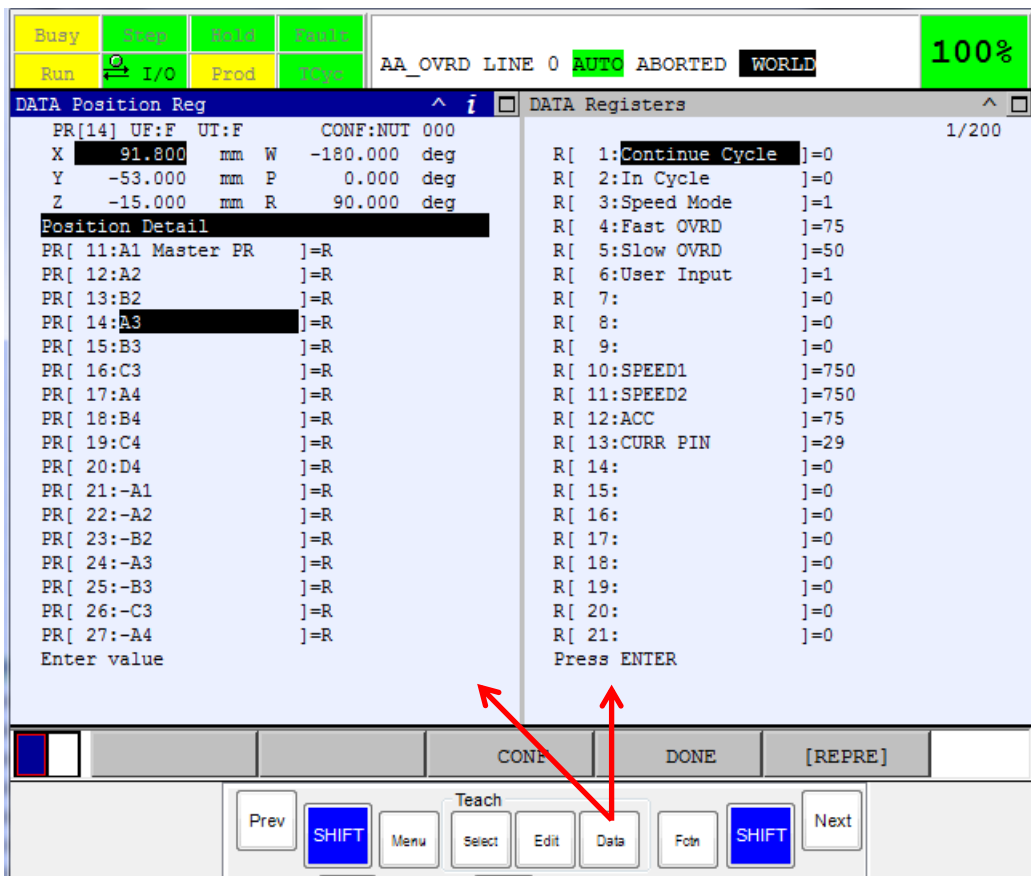
Register	Comment	Description	Default value
10 11 12	SPEED1 SPEED2 ACC	These are speeds and acceleration used for moves. They are set low to avoid frame shaking when on wheels. (If frame is fixed more securely, speed and acceleration could be increased)	750mm/sec 750mm/sec 75%
13	CURR PIN	This value is used internally by the program to keep track of the part number	n/a
16	IRV COUNT	Used to load 9 parts with Vision	n/a
100 101 102 103	ID_PICK_CYLINDER ID_PLACE_HOLE ID_REMOVE_CYLIND ID_EMPTY_PLATE	These values are used internally by the program to control the sequence of moves	n/a
105	Tmp_cyl_reg	This value is used internally by the program to set the status registers below	n/a
111 to 130	A1 in to -D4 in	These registers are used to store the status of the parts and holes. The values in these registers are linked to the Menu Displays using the iPendant Controls – see later section	1 = occupied 0 = unoccupied

5.2 Position Registers

Position Registers are used to store positions

Position Register	Comment	Description
11 to 30	A1 to -D4	Locations of the holes in the tray. Note that these are calculated values, not taught values. PR[11] was taught, then the other PR[]s were calculated relative to it.
31 to 39	Mid 1 to Mid 9	'Random' positions in middle of board – not in holes – robot will find actual position using <i>i</i> RVision
40 41	ToolOff ToolOff1	These are Tool Offsets, used to create the motion above / to the pick and place locations.

The Registers can be displayed using the 'Data' button on the Teach Pendant:



6 Program Details

6.1 List of programs

The following programs are installed in the Education Cell:

Program	Comment	Description
AAA_DEMO	Example program	This is Main Example program – it must be selected before pressing 'Cycle Start'
AA_CHK	Check Start OK	Program to check robot at home and parts in correct position. Uses Menu Utility to display choices and confirmation to user – see later section for details. Also uses .STM file to display graphic of correct part position – see later section for details.
AA_HOME	Move Home	This program moves the robot to the home position. Note that it uses Joint definition position – so independent of any UFrame or UTool settings
AA-OVRD	Set OVRD Fst Slo	Set override to value specified in R[4], R[5] depending on value of mode R[3]
ABORTIT	ABORT PRODUCTION	Predefined system program – not used by this application
DSP_WEBP	Display Web Page	Macro installed by the 'Menu Utility' Option. This macro is used to display the Menu Screens.
GETDATA	Get PC Data	Predefined system program – not used by this application
HAND_CLOSE	Close Gripper	Program to Close Schunk Gripper (Note 1)
HAND_OPEN	Open Gripper	Program to Open Schunk Gripper (Note 1)
HAND_TOG	Toggle Gripper	Program to toggle gripper between Open and Close. This program is linked to the User Key 1 on the Teach Pendant – see details in later section
IRV_LOAD	Get Parts Vision	Pick parts from centre of board using vision and place in left hand side of board

FANUC

Program	Comment	Description
IRV_PICK	Get 1 Part	Pick 1 part from centre of board using vision and place in left hand side of board – called from IRV_LOAD
IRV_RAND	Put Parts Vision	Pick parts from Right hand side of board and place in semi-random positions on middle of board for robot to find using iRvision
IRV_SET	IRV Setup Posn	Program containing suitable start position for iRvision Camera Calibration
LISTMENU	LIST MENU MACRO	Macro installed by the 'Menu Utility' Option. This macro is used to display 3 choices to the user when the robot is not at HOME
OPERMENU	Entry Menu Macro	Macro installed by the 'Menu Utility' Option. This macro is not used in this application
PROMPTOK	Prompt Box OK	Macro installed by the 'Menu Utility' Option. This macro is not used in this application
PROMPTYN	Prompt Box Y N	Macro installed by the 'Menu Utility' Option. This macro is used to confirm the choice by the user when the robot is not at HOME
REQMENU SENDDATA SENDEVNT SENDSYSV STATPAGE	Request PC Menu Send PC Data Send PC Event Send PC Sysvar StatusMenu Macro	Predefined system programs – not used by this application
S_SET1_PR	Solit Setup 1PR	Set coordinates of one Position Register for one hole
S_SETUP_PRS	Solit Setup PRs	Set coordinates of all Position Register for all holes
S_SOLIT_RESET	Reset Status Rs	Reset registers showing part status to start conditions
S_UNLD1	Solit Unload 1	Make one move, jumping over a piece and then moving the jumped-over piece to the other half of the board.
S_UNLDALL_L	Solit Unl All L	Sequence for the left side of the board, moving pieces to the right side

FANUC

Program	Comment	Description
S_UNLDALL_R	Solit Unl All R	Sequence for the right side of the board, moving pieces to the right side
S_UNLDLST	Solit Unld Last	Make one move – for first or last pieces
USERCLEAR USERPAGE	Clear User Page Show User Page	Macros installed by the 'Menu Utility' Option. These macros are not used in this application
ZERO	Move to Zero	Move all axes to zero

Note 1) In order for the Roboguide Simulation to work correctly picking and placing the virtual parts, the HAND_CLOSE and HAND_OPEN .TP programs must be replaced with Simulation programs. Please refer to actual Roboguide cell for details



6.2 Sample Program Listings

This is listing of main program "AAA_DEMO":

```
/PROG AAA_DEMO
  1: !FANUC EUROPE EDUCATION CELL ; << Remark
  2: !Example Solitaire Program ; << Remark
  3: !with optional vision ;
  4: CALL AA_CHK      ; << Call program to check Start OK
  5: R[1:Continue Cycle]=1    ; << Set Register to run continuously
  6: R[2:In Cycle]=1      ; << Set Register to show status
  7: CALL AA_OVRD      ; << Call program to set Override %
  8: CALL HAND_OPEN    ; << Make sure gripper is open
  9: LBL[1] ; << Label to loop up to if needed
10: CALL S_SOLIT_RESET  ; << Reset status registers
11: CALL S_UNLDALL_L    ; << Sequence to move pieces from Left to Right side of board
12: WAIT 2.00(sec) ; << Short delay
13: IF R[7:Vision Installed]<>1,JMP LBL[2] ; << Check if use Vision or not
14: IF R[8:Vision Enabled]<>1,JMP LBL[2] ; << Check if use Vision or not
15: CALL IRV_RAND      ; << Move parts from Right of board to Middle
16: WAIT 2.00(sec) ; << Short delay
17: CALL IRV_LOAD      ; << Pick parts from Middle of board using Vision and put in Left Side
18: WAIT 2.00(sec) ; << Short delay
19: JMP LBL[3] ; << Jump over next section
20: LBL[2] ; << Label for Jump
21: CALL S_UNLDALL_R    ; << Sequence to move pieces from Right to Left side of board
22: LBL[3] ; << Label for Jump
23: WAIT 2.00(sec) ; << Short delay
24: IF R[1:Continue Cycle]=1,JMP LBL[1] ; << Repeat if register 1 is equal to 1
25: R[2:In Cycle]=0    ; << Set register to show status at end of program
/POS
/END /END
```



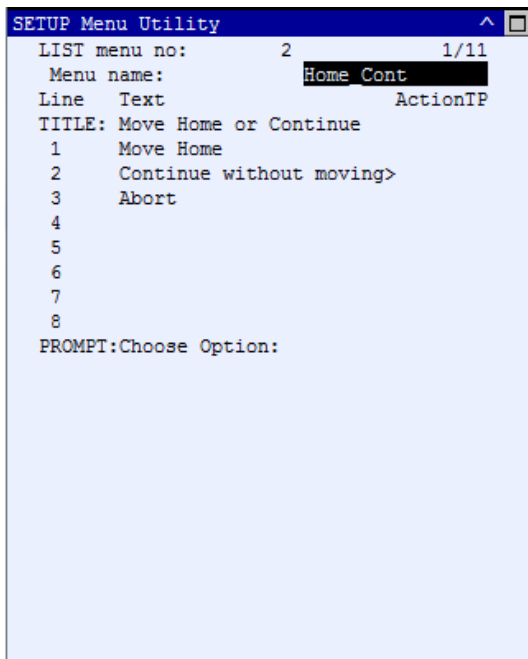
This is listing of program "AA_CHK":

```
/PROG AA_CHK
  1: !Check Start Conditions OK ;          << Remark
  2: ;
  3: !Check at HOME ;                    << Remark
  4: IF DO[101:HOME Signal]=ON,JMP LBL[10] ; << Check HOME signal - see later section for detail
  5: LBL[1] ;
  6: R[6:User Input]=0 ;
  7: CALL LISTMENU(2,6) ;                << Call Menu Utility to display User Menu 2, result in R[6]
  8: SELECT R[6:User Input]=1,JMP LBL[3] ;
  9:     =2,JMP LBL[5] ;
 10:     ELSE,JMP LBL[2] ;
 11: LBL[2] ;
 12: ABORT ;
 13: JMP LBL[10] ;
 14: LBL[3] ;
 15: CALL AA_HOME ;
 16: JMP LBL[10] ;
 17: LBL[5] ;
 18: R[6:User Input]=0 ;
 19: CALL PROMPTYN(2,6) ;                << Call Menu Utility to display Prompt Box 2, result in R[6]
 20: IF R[6:User Input]=1,JMP LBL[10] ;
 21: JMP LBL[1] ;
 22: LBL[10] ;
 23: ;
 24: !Check parts OK ;
 25: CALL S_SOLIT_RESET ;
 26: R[6:User Input]=0 ;
 27: !Display Check Page ;
 28: CALL DSP_WEBP(3) ;                  << Call Menu Utility to display User Status / Confirmation Menu
 29: WAIT R[6:User Input]<>0 ;
 30: IF R[6:User Input]=1,JMP LBL[20] ;
 31: ABORT ;
 32: LBL[20] ;
 33: !Display Run Page ;
 34: CALL DSP_WEBP(4) ; << Call Menu Utility to display User Status / Operation Menu
/END
```

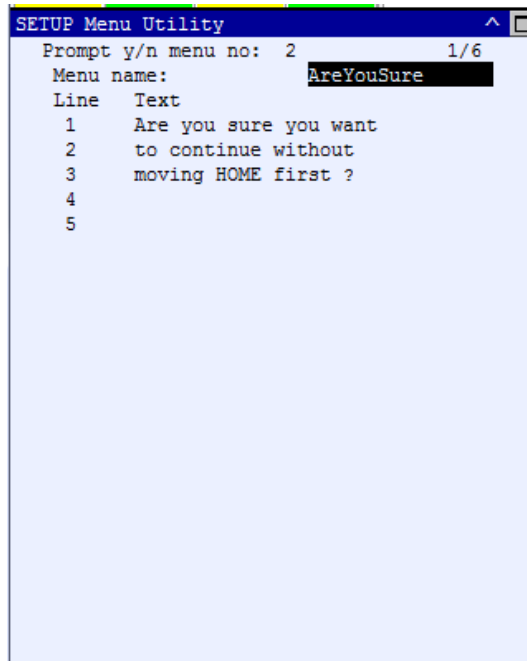
7 User Interface Setup

7.1 Menu Utility Setup

The User Menu and Prompt at the start of the AA_CHK program have been set up using the Menu Utility. See below for screenshots.



```
SETUP Menu Utility ^
LIST menu no:      2      1/11
Menu name:        Home Cont
Line  Text          ActionTP
TITLE: Move Home or Continue
1  Move Home
2  Continue without moving>
3  Abort
4
5
6
7
8
PROMPT:Choose Option:
```

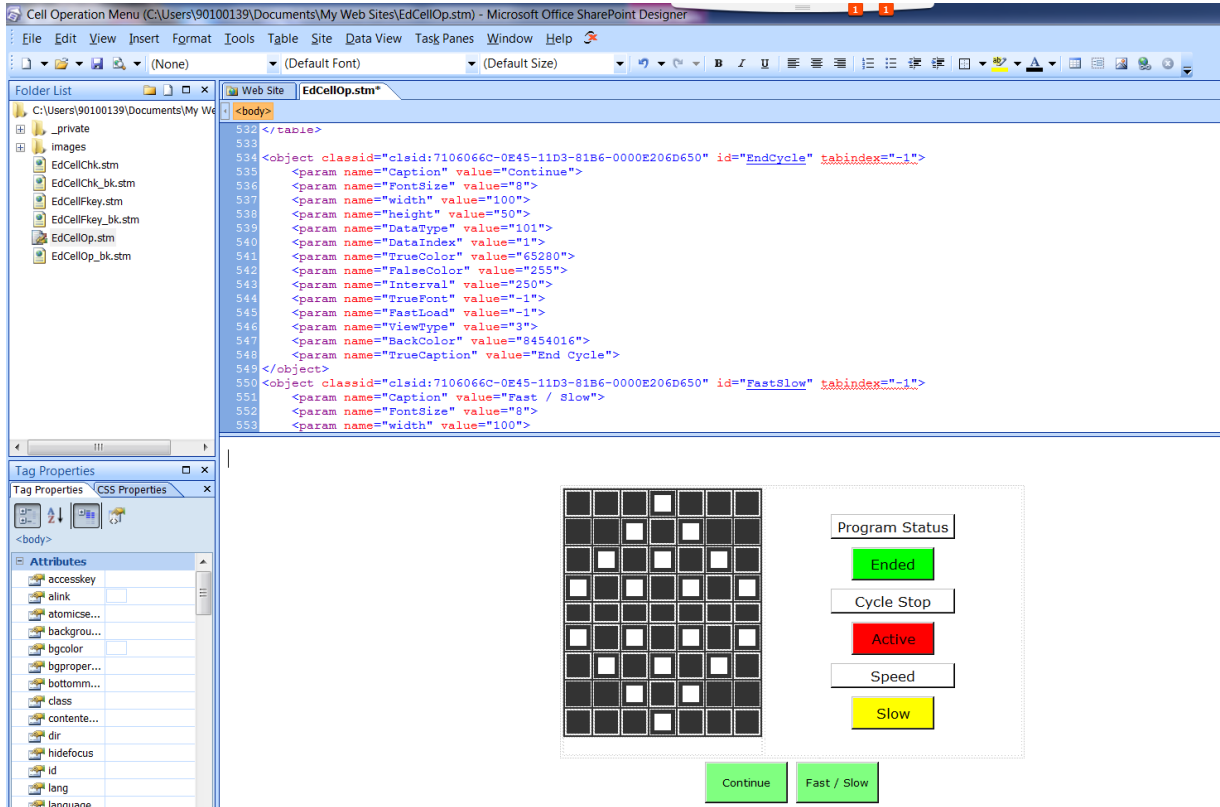


```
SETUP Menu Utility ^
Prompt y/n menu no: 2      1/6
Menu name:        AreYouSure
Line  Text
1  Are you sure you want
2  to continue without
3  moving HOME first ?
4
5
```

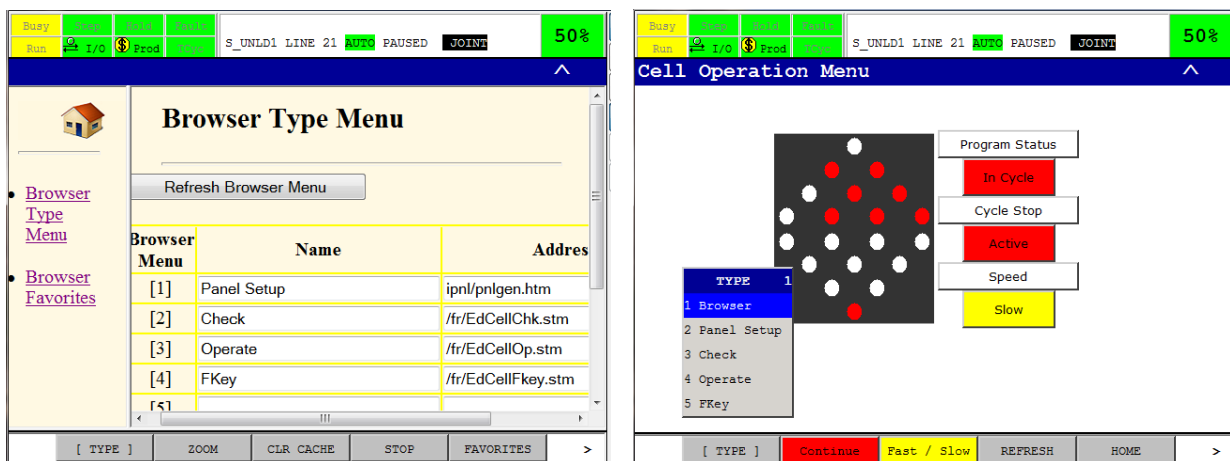
Once these menus have been set up, they can be called using the predefined macros "LISTMENU" and "PROMPTYN"

7.2 HTML Screens

The Status / Menu Screens were created using MS Sharepoint Designer 2007:



Then the files were loaded into the controller and added to the browser favourites menu:



So that they can be displayed manually or using the CALL DSP_WEBP(3) program, which is included in the Menu Utility

7.3 HTML Listing Extract

This is extract of listing for "EdCellOp.stm"

```
<head>           Title of the Page
<meta http-equiv="Content-Language" content="en-gb">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Education Cell Menu</title>
<style type="text/css">
.style1 {
    border-color: #FFFFFFF;
    border-width: 0;
    background-color: #333333;
}
.style2 {
    text-align: center;
}
</style>
</head>
<body>
<div class="style2">

<table style="width: 41%"> Table with two columns, left column for board layout, right column for
status boxes
    <tr>
        <td style="width: 235px">

<table style="width: 140px; height: 180px; float: right;" class="style1">
    <tr> Table with 7 columns and 9 rows to show board positions
        <td></td>
        <td></td>
        <td></td>
```

FANUC

<td style="width: 30px"> [Definition of one board position](#)

<object classid="clsid:71060668-0E45-11D3-81B6-0000E206D650" id="Sol1" style="width: 20px; height: 20px">

<param name="Caption" value="">

<param name="FontSize" value="14">

<param name="width" value="20">

<param name="height" value="20">

<param name="DataType" value="101"> [State linked to Register](#)

<param name="DataIndex" value="121"> [Register number 121](#)

<param name="TrueColor" value="255"> [True Colour = Red](#)

<param name="FalseColor" value="16777215"> [False colour = Light grey](#)

<param name="Interval" value="250">

<param name="TrueFont" value="-1">

<param name="FastLoad" value="-1">

<param name="Border" value="1">

<param name="ViewType" value="1">

</object>

</td>

<td></td>

<td></td>

<td style="width: 49px"></td>

</tr>

[Repeat for all other board positions up to 20](#)

</table> [Set up objects in right column](#)

</td> [Add label](#)

<td class="style2">

<object classid="clsid:7106065C-0E45-11D3-81B6-0000E206D650" id="FRIPLabel1" style="height: 30px; width: 150px">

<div class="style2">

<param name="Caption" value="Program Status">

FANUC

```
<param name="FontSize" value="10">  
<param name="width" value="150">  
<param name="height" value="30">  
<param name="DataType" value="100">  
<param name="DataIndex" value="">  
<param name="Interval" value="250">  
<param name="TrueFont" value="-1">  
<param name="FastLoad" value="-1">
```

```
</div>
```

```
</object>
```

```
<br> Add Lamp Object to show status of program
```

```
<object classid="clsid:71060668-0E45-11D3-81B6-0000E206D650"  
id="FRIPToggleLamp4" style="height: 40px">
```

```
<div class="style2">
```

```
<param name="Caption" value="Ended"> Set text for False
```

```
<param name="FontSize" value="10">
```

```
<param name="width" value="100">
```

```
<param name="height" value="40">
```

```
<param name="DataType" value="101"> Link to Register
```

```
<param name="DataIndex" value="2"> Register 2
```

```
<param name="TrueColor" value="255">
```

```
<param name="FalseColor" value="65280">
```

```
<param name="Interval" value="250">
```

```
<param name="TrueFont" value="-1">
```

```
<param name="FastLoad" value="-1">
```

```
<param name="TrueCaption" value="In Cycle"> Set Text for True
```

```
</div>
```

```
</object>
```

```
<br>
```

[Repeat for other lamps](#)

</table>

Add TP Key labels / functions objects

```
<object classid="clsid:7106066C-0E45-11D3-81B6-0000E206D650" id="EndCycle" tabindex="-1">
```

```
  <param name="Caption" value="Continue">
```

```
  <param name="FontSize" value="8">
```

```
  <param name="width" value="100">
```

```
  <param name="height" value="50">
```

```
  <param name="DataType" value="101"> Sets Register
```

```
  <param name="DataIndex" value="1"> Register 1
```

```
  <param name="TrueColor" value="65280">
```

```
  <param name="FalseColor" value="255">
```

```
  <param name="Interval" value="250">
```

```
  <param name="TrueFont" value="-1">
```

```
  <param name="FastLoad" value="-1">
```

```
  <param name="ViewType" value="3">
```

```
  <param name="BackColor" value="8454016">
```

```
  <param name="TrueCaption" value="End Cycle">
```

```
</object>
```

Repeat for other Key

```
</p>
```

```
<p>&nbsp;</p>
```

```
</div>
```

```
</body>
```

```
</html>
```


8 Technical Data

Technical data:	
Power rating	230V 16A single phase
Power consumption	1 KW/h
Connection	Schuko plug (German style)
Air supply	None
Air consumption	None
Installation size	Base 1,3 x 0,7 m Height 1,8m
Weight	170kg

Transport data	
Size	Base 1,3 x 0,7 m Height 1,8m
Weight	170kg
Packing	Bubble foil
Handling	On wheels