DVT- Fanuc Driver

DVT has now added new driver that implements Fanuc's Sensor Interface Protocol. This Protocol can be used to transfer information between Fanuc's Robot controllers (RJ-2/RJ-3) and other sensing devices such as DVT's SmartImage Sensor. The information is transferred through a point-to-point serial connection (RS-232C.) As with other drivers in the past, DVT has reserved a set of memory registers for Fanuc global variables. Using special function calls in DVT's script tool, information from the inspections (e.g. position of an object or number of objects found) is placed into the registers were the RJ-2/RJ-3 can access them with the SEND[] and REC[] Teach Pendant commands.

This document describes the details of the Fanuc-DVT interface. It includes the syntax of the functions needed to populate the registers in script. The document also shows how the physical connection is achieved and how the Fanuc driver is set and configured in FrameWork. Sample DVT scripts paired with portions of Fanuc Teach Pendant programs are shown to illustrate the process. A reference to the syntax of the functions can also be found in the FrameWork help files and in the Script tool Guide.

Basic Protocol Sequence

Fanuc's Sensor Interface Protocol requires a specific sequence of events. This sequence is shown in the figure below. Communications are initiated by the controller when a SEND[] instruction is encountered in the TPP (Teach Pendant). In this initial transaction, the controller sends a control register number (an integer from 0-255.) The control register number is the index or address of a given register in the controller's memory. This register has been previously set to a value different from 1 (usually 0 by convention.) From this point on, the controller is ready to receive general register data or position data from the sensor. The controller monitors the value of the control register until it becomes 1. This signals to the controller that the sensor has sent all its data and it ends the communication session. TPP Program execution is then resumed.

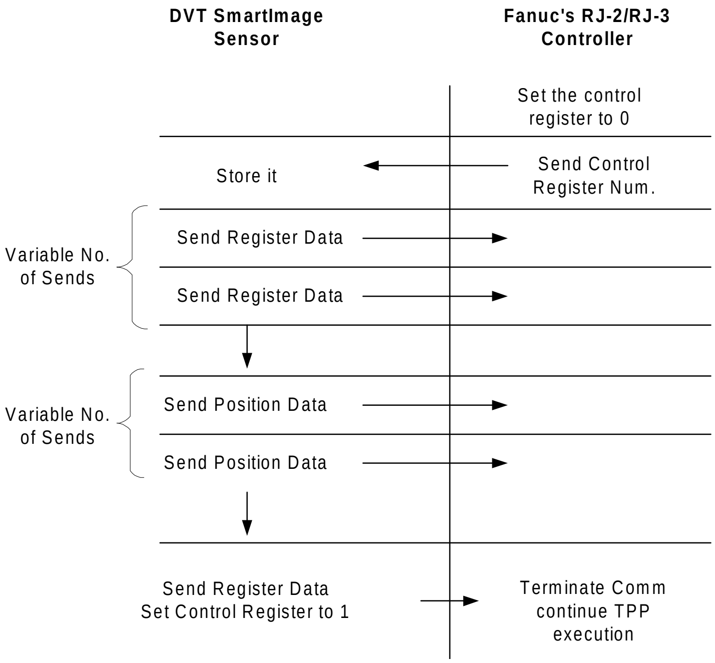

DVT SmartImage Sensor
Fanuc's RJ-2/RJ-3 Controller
Set the control register to 0
Store it
Send Control Register Num.
Variable No. of Sends
Send Register Data
Send Register Data
Variable No. of Sends
Send Position Data
Send Position Data
Send Register Data Set Control Register to 1
Terminate Comm continue TPP execution

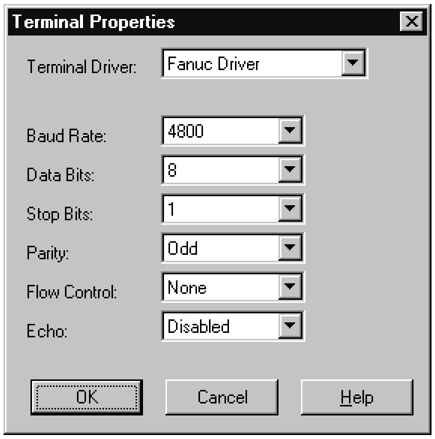
Figure 1: Sequence of Events in Fanuc's Sensor Interface Protocol

The protocol sequence as outlined above suggests that the sensor, in this case the DVT SmartImage Sensor, decides what and how much data is sent to the controller during each communication session. The Fanuc Sensor Interface Protocol does not allow the controller to make requests for specific data. In essence, all it does is provide a way for the sensor to signal the end of data transfer and then places the controller in a ready-to-receive-data state. The sensor must do the rest. Furthermore, the sensor has to decide were in the controller to place general register data; this information is part of the data packets. Position data is treated differently in that it is always placed in a position register defined by a system parameter in the controller (SENS_IF$.) This will be discussed in detail later.

As we will see, the structure of the Fanuc Sensor Interface Protocol requires that the DVT user write simple function calls inside a Script tool. These functions tell the Fanuc Driver what and how much information to send to the robot.

Configuring The Fanuc Driver in FrameWork

Install Framework 2.0 Beta 7 and download the firmware to the 600. After connecting, go to the **Comm.** menu and under **Image Sensor Communications.** Select the Fanuc driver and adjust the parameters as shown in the following window. These parameters are recommended by Fanuc's documentation.



Terminal Driver: Fanuc Driver

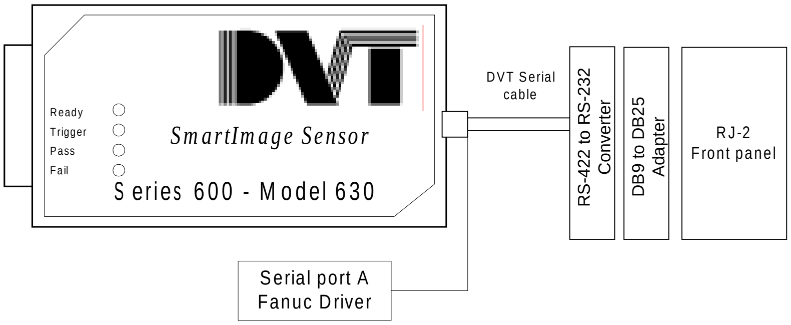
Baud Rate: 4800
Data Bits: 8
Stop Bits: 1
Parity: Odd
Flow Control: None
Echo: Disabled

Figure 2 : Fanuc Driver Configuration in FrameWork

After configuring the driver, a product must be created. This product must contain at least one script tool that properly uses the Register Management functions for The Fanuc driver. At least one inspection must be triggered so that the values are actually written to the registers. When the controller starts communications the driver will now respond properly via the serial port.

Physical Connection

The RJ-2 controller works with the DVT standard self-powered converter from BlackBox (DVT Part Number: CON-142). The RJ-2 supports pins in the RS-232 which allow the converter to draw power. A schematic of the connection is shown below. Make sure the DB9 to DB25 adapter is straight through, not a null modem.



The pin out of the RJ-2's serial connector located in the front of the unit is shown below.

Pin No.	Function
2	TxD
3	RxD
4	RTS
5	CTS
6	DSR
20	DTR

Configuring The Sensor Interface Protocol in the Fanuc Controller

There are several configuration steps required in the Fanuc controller to ensure proper communications.

1. Make sure that the controller has the proper firmware options installed. The Sensor Interface protocol is not standard. It is an option that must be obtained separately. Contact Fanuc's Tech support for details. 1800-47-ROBOT.

2. Set up the serial port. This is done by editing the system parameter $RS_232. Alternatively this can easily be done with the TPP (Teach Pendant) by going through **Menus...Setup…Port Init**. The **Device** must be set to **Sensor** (this enables the Sensor Interface protocol.) The other communication parameters must match those set previously in FrameWork's Fanuc driver. Use parameters recommended by Fanuc: 4800, odd parity, 1 stop bit.

TPP Programming and the Sensor Interface

This section is largely extracted from Fanuc's documentation. There are three TPP program instructions provided by the FANUC Sensor Interface:

- `SEND R[*]`
- `RECV R[*], LBL[*]`
- `RECV R[*] LBL[x] TIMEOUT LBL[y]`

In the following sample code,

·R[1] is set to the number of the register to be used for comm. control (Control Register).

·R[2] must be set to 0 (R[2] is the control register in this case because R[1] =2 is sent.)

·SEND R[1] sends a packet (the value 2) from the robot to the Sensor.

·The RECV command waits for R[2] to become non-zero. The Sensor can send values to any number of General Registers during this time. The Sensor can also send a Position to a Position Register.

·The Position Register which will receive the value is set by the system variable $SENS_IF[1].

·The OFFSET CONDITION indicates the position register which has received the value from the Sensor (i.e., PR[$SENS_IF[1]].) Ensure that your taught position (P[1]) includes the proper configuration.

Sample TPP Program:

```
R[1] = 2
R[2] = 0

! Notify the sensor to use R[2] to terminate comm..
SEND R[1]

! Jump to label LBL[1] in case of comm. error
RCV R[2], LBL[1]

! Assume $SENS_IF[1] = 4
OFFSET CONDITION PR[4]
J  P[1] 100% FINE OFFSET
```

Register Management Functions for the Fanuc Driver

Fanuc Registers in DVT memory

Fanuc Data in DVT registers						
Fanuc Types	**Size in DVT memory**	**Variable Number**	**# of DVT Registers**	**Description**	**Fanuc's Range**	**Units**
General Register	4 byte	0-255	4	Integer - Signed	+ 8388607 to –8388608	
Position Register	24 byte	0 to 10	24	Integer- Signed	+ 83886.07 to – 83886.08 mm + 83.8607 to – 83.88608 rad	mm/rads

FanucSetup()

Syntax:

```
FanucSetup(RegsToSend as Integer, PosRegsToSend as Integer)
```

Arguments:

RegsToSend: This argument specifies the number of general registers to send during the communication session. The valid range is **0-255**. The range is defined by the number of general registers in the Fanuc RJ-2 controller.

PosRegsToSend: This argument specifies the number of position registers to send during the communication session. The valid range is **0-10**.

Example:

`FanucSetup(1,1)->` Tells driver to send 1 data register and 1 position register. The contents of general register 1 in the DVT sensor are sent to the controller's general register 1. Position register 1 in the DVT sensor is then sent to the position register indicated by the SENS_IF$ parameter in the controller.

Notes:

Registers are sent in ascending order only. For instance: `FanucSetup(5,0)` will cause the driver to send general registers 1-5.

FanucWriteReg()

Syntax:

```
FanucWriteReg(RegNum as Integer, value as Integer)
```

Arguments:

RegNum: This argument specifies the address of the Fanuc general register where *value* will be placed. It also specifies the address of the general register in the controller where the data will be sent during the communication session. The valid range is **0-255**. The range is defined by the number of general registers in the Fanuc RJ-2 controller.

value: This argument specifies the data. It may be a literal number or a value from the sensors (see examples below.) The valid range is **+ 8388607 to -8388608**. Internally, the Fanuc Controller stores general registers in signed 3-byte numbers. This is why the range is smaller than the valid range for a DVT 4-byte Integer.

Example:

`FanucWriteReg(1,1267)->` Writes 1267 to Fanuc's general register #1 in DVT's registers. During the comm. session, the value 1267 is written to fanuc's general register #1 on the RJ-2 controller.

Notes:

FanucWritePReg()

Syntax:

```
FanucWritePReg(PRegNum as Integer, X as Double, Y as
               Double, Z as Double, Tx as Double, Ty as
               Double, Tz as Double)
```

Arguments:

PRegNum: This argument specifies the address of the Fanuc position register where the data will be placed. The Valid range is **0-10**.

X,Y,Z: These arguments specify the position data. They may be literals or values from the sensors. The units for this values must be **mm**. The valid range for positions is **+ 83886.07 mm to –83886.08 mm**. Internally, the Fanuc Controller stores each element of a position register as a scaled, signed, 3-byte number. This explains the range and units requirement.

Tx,Ty,Tz: These arguments specify the rotation. They may be literals or values from the sensors. The units for these values are radians. The valid range for these angles is **+ 83.8607 rad to –83.88608 rad**. Internally, the Fanuc Controller stores each element of a position register as a scaled, signed, 3-byte number. This explains the range and units requirement.

Example:

```
FanucWritePReg(1,1.0,2.0,0,0,0,PI/4)

FanucWritePReg(1, sel.BlobTransformedPoint.X[1],
               sel.BlobTransformedPoint.Y[1],0,0,0
               (SoftSensor.BlobAngle[1])*(PI/180))->
```

Writes the position to Fanuc's Position register #1 in DVT's registers. During the comm. session, it is sent to a position register defined by $SENS_IS on the RJ-2 controller.

Notes:

Z, Tx, and Ty are currently forced to zero. Remember the different formats for angle output in DVT.

Examples

In the Following section contain examples of DVT scripts paired with corresponding TPP programs.

Example 1: Transferring General Register Data from DVT to a Fanuc Controller

Suppose we need to tell the robot how many objects of type 1 and type 2 were found after an inspection. This requires transmission of two general registers.

DVT script: must be executed at least once.

```
int type1;
int type2;

type1=seltype1.NumBlobs;
type2=seltype2.NumBlobs;

FanucSetup(2,0);  //Sets up to send general regs 1 and 2

FanucWriteReg(1,type1);//Writes inspection values into regs
FanucWriteReg(2,type2);
```

Fanuc TPP Program: Here we only show the portion relevant to comm. is shown.

```
R[10] =11
R[11] = 0

!Notify the sensor to use R[11] to terminate comm..
SEND R[10]

! After this instruction the controller receives data
! Until R[11] is set to 1, then continues TPP exec.

RCV R[11], LBL[1]

! Continue execution of TPP PRGRAM. R[1], R[2] now have values from DVT
SmartImage Sensor
```

Example 2: Transferring Position Data from DVT to a Fanuc Controller

Suppose we need to tell the robot the position and orientation of an object.

DVT script: must be executed at least once.

```
Double PI;

PI=3.14159;
FanucSetup(0,1); //Set up to send the 1st pos register.

FanucWritePReg(1, sel.BlobTransformedPoint.X[1],
sel.BlobTransformedPoint.Y[1],0,0,0
(SoftSensor.BlobAngle[1])*(PI/180));
```

Keep in mind that the values from the sensors should be in the allowable ranges for the functions.

Fanuc TPP Program: Here we only show the portion relevant to comm. is shown.

```
R[10] =11
R[11] = 0

! Notify the sensor to use R[11] to terminate comm..
SEND R[10]

! After this instruction the controller receives data
! Until R[11] is set to 1, then continues TPP exec.

RCV R[11], LBL[1]

! Continue execution of TPP. Perform shift of taught position to pick the part

! Assume $SENS_IF[1] = 4
OFFSET CONDITION PR[4]
J  P[1] 100% FINE OFFSET
```