IAI – DVT SENSOR/VISION INTERFACE

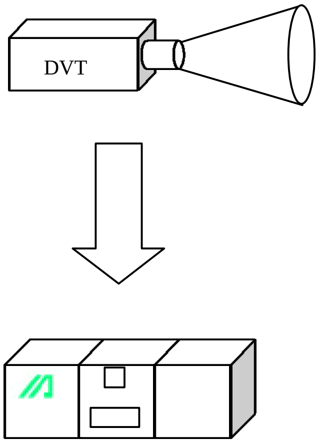
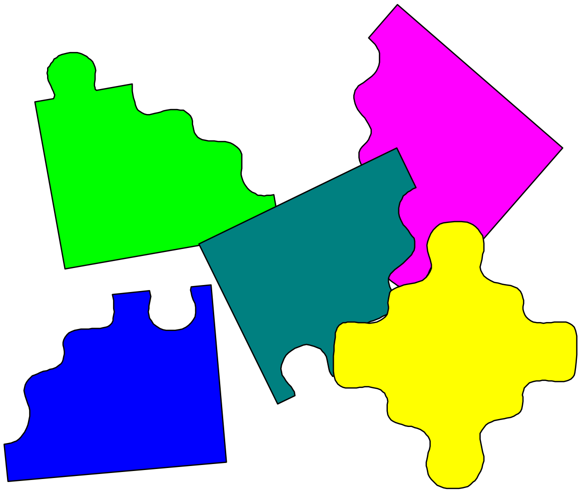


TABLE OF CONTENTS

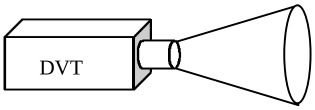
INTERFACING VISION SYSTEMS WITH THE IAI CONTROLLER CAN SEEM INTIMIDATING AND CONFUSING.
HOWEVER, WHEN BROKEN DOWN INTO SMALLER COMPONENTS, IT'S LIKE PUTTING TOGETHER PIECES OF A PUZZLE



A complete vision guided motion application contains the following major components:

- **Setup the product or products in the vision system**
- **Devise and implement the communications between the IAI and vision system**
- **Program any necessary math conversion formulas**
- **Program motion portion of the IAI system**
- **Configure I/O interface with external devices (i.e. end effectors, PLC's, etc)**

CONFIGURING PRODUCTS ON THE DVT



- The DVT has a simple, easy to use PC interface software package called FrameWork

- Product configuration is determined by the use of DVT's SoftSensors

- Various SoftSensors give the ability to measure object position, rotation, and intensity

- After an initial image is obtained, the Framework software is used to easily placed on the object

- Some fine-tuning is often required to ensure that the SoftSensors are placed optimally

- Converting from the DVT pixel information to the real world is achieved by a math SoftSensor

- Simply choose two distinctive points on the object and obtain both pixel and IAI coordinates for each point using a combination of FrameWork and SELWIN

- This completes the product setup for the vision system

- Repeat these steps for all desired objects for the application

SERIAL COMMUNICATIONS BETWEEN THE DVT AND THE IAI

- **The serial string to communicate the data from the DVT to the IAI is easily setup within the FrameWork software**

- **The string is created from data that results from the SoftSensors**

- **For this application, the following string communicated the X, Y, and Theta data:**

XXX.XXXYYY.YYYTTT.TTT{CR}{LF}

- **This string is interpretted through the RS232C 2 channel option (SELNET) on the IH controller**

- **The FrameWork software can easily construct a string for the teach port to read if desired**

- **The serial cable itself requires only transmit, receive, signal ground and a shield to transfer the RS232 signal**

- **The IAI controller reads the string and then uses the SELNET commands to convert the data from ASCII caharacters to variables**

- **Once the data from the vision system is in variable form inside the SEL controller, the communications section is complete**

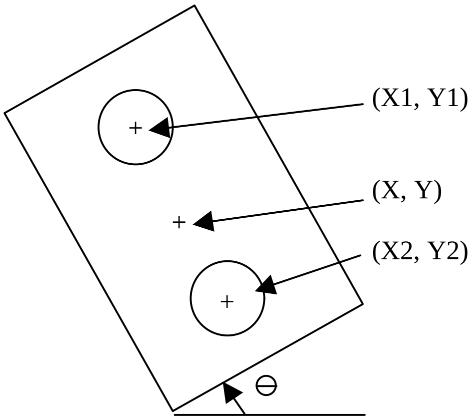
MATH CONVERSION

- **If your system has any rotation or offsets involved, it may be necessary to convert the vision data before moving to the positions**

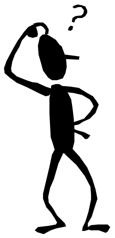
- **For this SCARA example, the card could be at an angle, the end effector is 33mm from the center of the vertical shaft, and the pucks are placed 22mm from the center of the card**

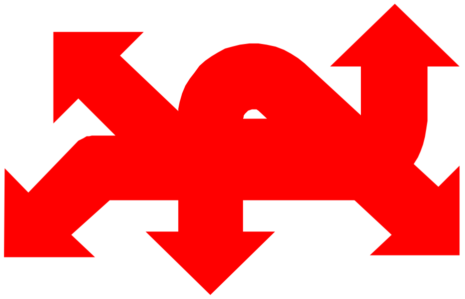
- **The following equations and diagram show the final equations to convert the X and Y position data:**

X1 = X – 33COS(Theta) – 22SIN(Theta)
Y1 = Y – 33SIN(Theta) + 22COS(Theta)
X2 = X – 33COS(Theta) + 22SIN(Theta)
Y2 = Y – 33SIN(Theta) – 22COS(Theta)



- **This part can be confusing!**





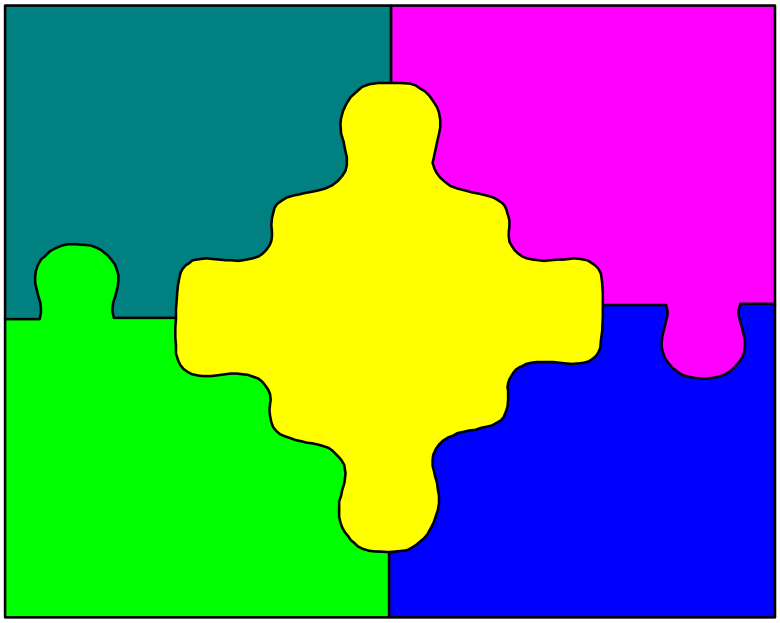
FINISHING TOUCHES

- Once the vision data has made its way into the IAI controller and been converted, the remainder of the application is really no differeent than typical IAI applications

- In essence, the vision system is nothing more than a high tech method of entering data into the point table

- DVT offers other options including I/O to trigger an image acquisition or select one of several possible products stored in the DVT system

- At this point the application is complete!



IAI – DVT INTERFACE – TECHNICAL DISCUSSION

PURPOSE

IAI controllers of various types can interface with vision systems either through serial communications or discreet I/O. Discreet I/O interface with vision systems is quite limited allowing only the most basic of information to be conveyed (i.e. good part/bad part decision-making). Therefore, we have developed this information package based on serial communications between the IAI and a vision sensor. Ideally, this discussion will give you the basic knowledge of how IAI communicates serially with other devices – specifically with the DVT system. In order to fully appreciate the interface, the discussion shall also cover many details pertaining to the DVT system itself. Keep in mind that IAI has successfully interfaced with DVT, Cognex, Omron, and Keyence vision systems to name a few.

DVT BACKGROUND

DVT offers several lines of “vision” products including the 600, 700, and 800 series. Their products are frequently referred to as smart sensors, not vision systems. They offer an inexpensive system that does not provide all the capabilities as a complete package system such as Cognex. Therefore, not all applications can be solved with a DVT system. This is application specific and the customer must determine the vision system required for their needs. The 600 series is the latest and greatest and offers increased speed, resolution, etc over the 700 and 800 series. For this demonstration, we used the 700 series due to its availability.

The DVT software – FrameWork – offers several “SoftSensor” options (see attachment). These softsensors allow the user to determine information such as object position (via pixel on/off mode), rotation, numbers of edges or features, intensity, measurement/math, or blob analysis. For the majority of motion guidance applications, the user needs only a few of these tools. The software itself is easy to use if a bit simplistic. Basic setup involves obtaining an ideal image and placing the appropriate SoftSensors on the image. These SoftSensors and their placement on the object define the “product.” The product is then saved to the DVT board itself. The DVT board can store up to 16 different products.

DVT also offers on-board 24VDC I/O to interface with the IAI controller. Inputs may be used to select which of the 16 products to look for and when to look for them. Outputs can indicate pass/fail information for an individual sensor or the entire object.

Serially, the DVT system offers a simple, easy to use setup where the user can dictate the data that is sent in the string and which format this data will be presented. This allows the user to setup whichever protocol they would like to use, thus allowing easy implementation of IAI teaching port communications.

WIRING

The DVT may communicate with the IAI through either SELNET, RS232C 2 channel option, or the teach port. Each cable type pinout is shown below.

DVT	**SELNET**	**2 Channel RS232C**	**Teach Port**
(9 Pin Female)	**(6 Pin Female)**	**(9 Pin Female)**	**(25 Pin Female)**
1			
2 (TD)	3 (RD)	2 (RD)	3 (RD)
3 (RD)	4 (TD)	3 (TD)	2 (TD)
4			
5 (SG)	5 (SG)	5 (SG)	7 (SG)
6			
7			
8			
9			

EXAMPLE

Although the available demo, lens used and implementation materials limit this example, it effectively demonstrates a medium difficulty vision application interfacing a DVT system with an IAI SCARA robot. The objective is to place two metal pucks from a known location onto a randomly (in both position and rotation) placed business card. Given the objective, we know several requirements for the DVT system. First we must ensure that the card is fully within the field of view. Second, we must find the center point of the card. Third, we must determine the angle that the card is placed at.

In order to make the math slightly easier, the SCARA brings the camera in-line with the Cartesian coordinate system of the SCARA before obtaining the image. If the camera were skewed with respect to the coordinate system, additional complexity would be added to the setup. Therefore, if the user cannot mount the camera independent from the SCARA itself, care should be taken to ensure positional orthogonality before proceeding.

To accomplish these tasks, we first need to use an edge detector sensor to determine if the card is fully within the field of view. We then use an X-Y translational sensor to locate the center of the card. However, this information is provided in pixels and must be converted. Hence, we use a math tool to transform from pixel coordinates to real world SCARA coordinates. This is easily accomplished by obtaining an image, finding two distinctive points on the image, and recording the position of these points both in pixels (using the FrameWork software) and in SCARA coordinates (using the SELWIN software). The final task of determining the rotation is the most difficult. In order to accomplish the feat, we must attach a sensor's location to the result of another sensor. We must find the vertical location of middle of the card, then find the right-most edge along the middle of the card, and finally find the angle at the right side of the card. Therefore, we use two translational sensors and a rotational sensor that are all interconnected.

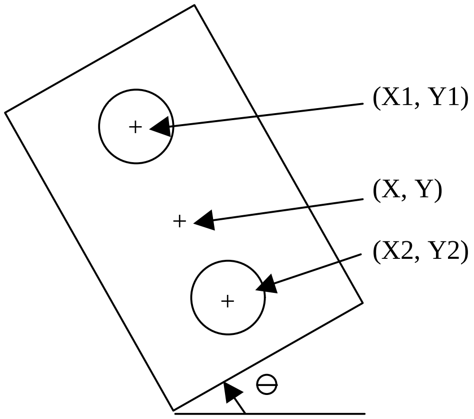
Once the DVT obtains this data it is sent back as a string in the following format:

XXX.XXXYYY.YYYTTT.TTT{CR}{LF}

where we read X, Y and Theta data respectively. We interpret this string through the 2 Channel RS232C option (SELNET). Due to its simplicity it is a fairly easy SEL issue to interpret the string and write the data to variables.

Once the data is in variables, complicated trigonometric functions are required to adjust for the tooling offset and the placement offset for each puck. If we consider the information we receive from the DVT as X, Y, Theta, the following equations will convert to the actual position data required.

X1 = X – 33COS(Theta) – 22SIN(Theta)
Y1 = Y – 33SIN(Theta) + 22COS(Theta)
X2 = X – 33COS(Theta) + 22SIN(Theta)
Y2 = Y – 33SIN(Theta) – 22COS(Theta)



After we calculate the final position data, the controller requires a simple pick and place motion sequence to place the pucks on the card and then reverse that sequence to retrieve them. The program contains no real complexity outside of the math required to convert position data based on rotational offsets. Even the math, while somewhat complicated, is not terribly difficult. Hence, this application came together quite quickly considering we started with almost zero knowledge of the DVT system to start with and we made it more complex than necessary (to show how a typical real-world application would actually progress). Therefore, the DVT system has shown that, while limited in certain areas, is relatively easy to setup and interface with the IAI controller. Portions of this exercise are generic and can therefore be used for any DVT – IAI integration (such as the communication strings, SELNET portion, and wiring setup). Other portions are application specific (such as the math functions, motion portion, and the DVT sensor setup) requiring the customer to write these portions of the code.

In the near future, we hope to develop more standardized interfaces with DVT and all other major vision and HMI packages. The goal is to make the interface package as simple and generic as possible for each system in order to facillitat our user's implementation of these systems. As we've seen, these types of interfaces are becoming more and more important in today's marketplace, and we need to be capable of providing easy solutions to our customer's needs.

SELNET – DVT SuperSEL Program

```
1          VEL   100
2          VELS  30
3          ACCS  20
4          SCHA  10                LF = ENDING CHAR
5          MOVP  446
6          MOVP  447
7          TAG   1
8          TIMW  0.5
9          WTON  10                CYCLE START
10         OPEN  1
11         READ  1     1           READ DVT DATA
12         CLOS  1
13         ISNE  15    '#'         # = ERROR
14         SLEN  7
15         VAL   190   1           VAR 190 = Y DATA
16         VAL   191   8           VAR 191 = X DATA
17         VAL   199   15          VAR 199 = THETA
18         PPUT  4     450         SET THETA FOR T1
19         PPUT  4     449         SET THETA FOR T2
20         CPLE  190   10    910   ERROR CHECK
21   910   GOTO  2
22         EXSR  1                 TRIG
23         JUMP  438               PICK PUCK 1
24         BTON  303               MAGNET ON
25         TIMW  0.5
26         JUMP  450               PLACE PUCK PT 1
27         BTOF  303               MAG OFF
28         TIMW  0.5
29         JUMP  439               PICK PUCK 2
30         BTON  303               MAG ON
31         TIMW  0.5
32         JUMP  449               PLACE PUCK PT 2
33         BTOF  303               MAG OFF
34         TIMW  0.5
35         JUMP  438               PARK POS
36         WTON  11
37         JUMP  450               REPLACE PUCK 1
38         BTON  303               MAG ON
39         TIMW  0.5
40         JUMP  438
41         BTOF  303               MAG OFF
42         TIMW  0.5
43         JUMP  449               REPLACE PUCK 2
44         BTON  303               MAG ON
45         TIMW  0.5
46         JUMP  439
47         BTOF  303               MAG OFF
```

```
48    TIMW  0.5
49    JUMP  445                  CAMERA READ POS
50    TAG   2
51    EDIF
52    GOTO  1
53
54    BGSR  1                    TRIG CONVERSION
55    MULT  199    0.01745       CONV TO RADIANS
56    SIN   192    *199
57    COS   193    *199
58    LET   194    *192
59    MULT  192    22
60    MULT  194    33
61    LET   195    *193
62    MULT  193    22
63    MULT  195    33
64    LET   199    *191
65    SUB   199    *195
66    SUB   199    *192
67    PPUT  1      450           SAVE X1
68    LET   199    *191
69    SUB   199    *195
70    ADD   199    *192
71    PPUT  1      449           SAVE X2
72    LET   199    *190
73    SUB   199    *194
74    ADD   199    *193
75    PPUT  2      450           SAVE Y1
76    LET   199    *190
77    SUB   199    *194
78    SUB   199    *193
79    PPUT  2      449           SAVE Y2
80    EDSR
```