



Applied Systems Engineering, Inc.

Technical Note #31 IEC Controls

Technical Note 31: IEC Controls

IEC Controls.....	3
Locating the Control Variable.....	3
Writing the Control.....	4
Control Terminations Events.....	5
Example Code	6

IEC Controls

To keep things simple, let's consider an IEC device that only supports one IEC information type (Single Controls). This device might look like the following:

DAC Offset	IEC Object Information Address
0	5000
1	5005

The device contains two control points with IEC Object Information Address 5000 and 5005 respectively. Now assume that the remote station wants to control the point with Object Information Address 5005. The remote station transmits ASDU 45 with the following parameters Type = 45, Address = 5005, Function = Select, Value = 1, and COT = ACTIVATION. The IEC performs the following steps to perform the control.

Locating the Control Variable.

Before the actual control can be performed, the IECPT must determine if the user supports the variable. The IECPT calls the DAC Find API on the Single Bit Command Object to locate the variable. The call is made as follows:

```
rc = (*DacObjects[USER_IEC_SINGLE_COMMAND].Find)(,,Id = 5005, DACOBJREF *off );
```

The find API is called to locate the DAC control variable. The Id specified in the call is 5005, the Object Information Address of the variable provided by the remote station. The result returned by the user application determines whether the variable exists. The following table provides the meaning of the return code.

DAC Find API return codes	Description
<= 0	The specified variable does not exist
=1	The specified variable does exist. The offset returned by the user determines the position of the control variable in the array of all control variables supported by the object.

In the above example if the DAC Find API returns a value <= 0 the control requested is rejected by the IECPT (COT=ACTCON, P/N = 1). If the value returned by the find API is 1 the control variable is supported by the user. In addition the offset returned by the find is the position that the control variable occupies in the array of all control variables. In the example above the offset would be 1. The IEC control variable Id = 5005 is located at offset 1 in the variable array.

Writing the Control

After locating the control variable, the IECPT next writes the parameters passed by the remote station to the DAC variable. The write is performed using the DAC Write API as follows:

```
rc = (*DacObjects[USER_IEC_SINGLE_COMMAND].Write)(,,,Start=1,End=1,Params );
```

The start and end offsets passed to the write API is the offset value returned from the DAC Find API (see previous section). The control parameters are passed in a BINARYOUTPUT structure. They are parsed from the ASDU transmitted by the remote station. The setup of the BINARYOUTPUT structure is as follows:

Field	Setting/value
Id	Object Information Address (5005)
Function	One of the following: DAC_FNC_SELECT DAC_FNC_EXECUTE DAC_FNC_CANCEL
Value.Variation	One of the following: DAC_VAR_BO_SINGLE DAC_VAR_BO_DOUBLE DAC_VAR_BO_BIT_STRING
Value.Clear	Not used.
Value.Bo.Precision	32 for bit string outputs.
Value.Bo.u8	Control value. Supplied in remote station request or DAC control event.
Value.Ontime.Tag	DAC_BO_DURATION_QUALITY
Value.Ontime.Duration.Quality	One of the following: DAC_BO_OUTPUT_NO_INFO DAC_BO_OUTPUT_SHORT DAC_BO_OUTPUT_LONG DAC_BO_OUTPUT_PERSISTENT

The user application performs the request control operation. The return code from the DAC Write API and the result field in the BINARYOUTPUT structure determine the result of the control. The return code from the DAC Write API is used by the IECPT as follows:

DAC Write API return codes	Description
< 0	The control operation failed
> 0	The control operation completed successfully. The result field in the BINARYOUTPUT structure contains the result of the control.

The result field is set by the user application. It can take on the following values.

Field	Setting/value
Result	One of the following: 0, Success

Technical Note 31: IEC Controls

	DAC_RESULT_ACTCON_OK DAC_RESULT_ACTCON_FAILURE
--	---

If the return from the DAC Write API is > 0 (control completed) the result field in the BINARYOUTPUT structure is used to build the appropriate response ASDU transmitted back to the remote station.

Control Terminations Events

The user application must provide a completion event when the control operation terminates. The IECPT does not build a response ASDU for the control. Instead it waits until a control completion event (BINARYOUTPUTEVENT) is received from the user application.

When the control does complete the user application inserts into the event queue a BINARYOUTPUTEVENT. To create a termination event the user application copies all field passed to the user application in the DAC Write API BINARYOUTPUT structure into the BINARYOUTPUTEVENT structure. In addition the user application sets the result field to one of the results described below.

Field	Setting/value
Result	One of the following: 0, Success DAC_RESULT_ACTTERN_OK DAC_RESULT_ACTTERM_FAILURE

The BINARYOUTPUTEVENT is then inserted into the event queue (high priority). When the IECPT reads the BINARYOUTPUTEVENT from the queue the control termination event is transmitted to the remote station.

Control Example

The following table goes through all of the steps in controlling IEC variable 5005.

Step	Remote	IECPT/DAC
1	Remote station sends ASDU (Type = 45, COT=ACTIVATE, FNC=SELECT, VALUE=1,DURATION=PERSISTANT)	
2		ASDU is parsed. DAC Find API is called with ID = 5005. Return Code <=0 IEC sends ASDU (Type = 45, COT=ACTCON, P/N = 1). Return Code = 1, Offset = 1 Goto Step 3
3		IEC calls DAC Write API with BINARYOUTPUT and start = end = 1.

Technical Note 31: IEC Controls

		<p>Return Code < 0 IEC sends ASDU (Type = 45, COT=ACTCON, P/N = 1).</p> <p>Return Code > 0. IEC sends ASDU (Type = 45, COT=ACTCON).</p>
4	Remote station sends ASDU (Type = 45, COT=ACTIVATE, FNC=EXECUTE, VALUE=1, DURATION=PERSISTANT)	
5		<p>ASDU is parsed. DAC Find API is called with ID = 5005.</p> <p>Return Code <= 0 IEC sends ASDU (Type = 45, COT=ACTCON, P/N = 1).</p> <p>Return Code = 1, Offset = 1 Goto Step 6</p>
6		<p>IEC calls DAC Write API with BINARYOUTPUT and start = end = 1.</p> <p>Return Code < 0 IEC sends ASDU (Type = 45, COT=ACTCON, P/N = 1).</p> <p>Return Code > 0. IEC sends ASDU (Type = 45, COT=ACTCON).</p> <p>Goto Step 8</p>
7		<p>IECPT receives termination event. When event received.</p> <p>Result = ACTTERM_OK IEC sends ASDU (Type = 45, COT=ACTTERM).</p> <p>Result = ACTTERM_FAIL IEC sends ASDU (Type = 45, COT=ACTTERM, P/N = 1).</p>

Example Code

```

BINARYOUTPUT  save;

DACAPI DacWriteBinaryOutput( , , LPBINARYINOUT p )
{
    if ( p->Function == DAC_FNC_EXECUTE )
    {
        user performs control operation;
        memcpy( save, p, sizeof(BINARYOUTPUT) );
    }
}

/* Called by user code when the control execution completes or fails due to timeout or
other error */
DACAPI UserControlCompletion( BOOL success )
{
    BINARYOUTPUTEVENT  b;

    b.Id = save.Id;
    memcpy( b.Value, save.Value, sizeof(BOVALUE) );
}

```

Technical Note 31: IEC Controls

```
b.Result = (success) ? DAC_RESULT_ACTTERM_OK : DAC_RESULT_ACTTERM_FAILURE;  
  
Insert BinaryOutputEvent into event queue (priority high to insert event at the  
Front of the queue.  
}
```