



EPCIO Series Device Driver Library Example Manual

Version: V.2.01

Date: 2008.07

<http://www.epcio.com.tw>

Table of Contents

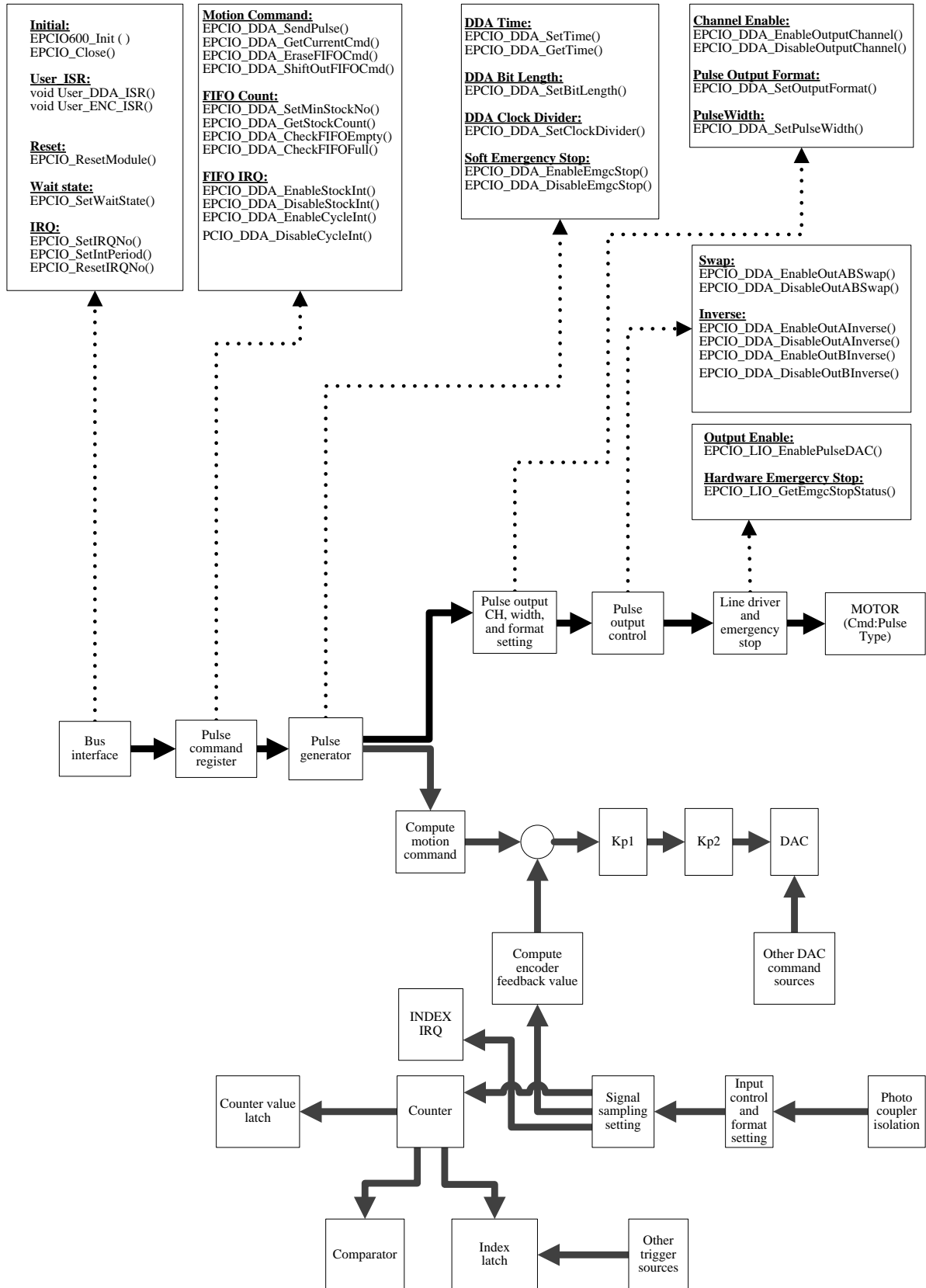
| | |
|--|-----------|
| 1. DEVICE DRIVER CALLING FLOWCHARTS | 3 |
| 1.1 DDA | 4 |
| 1.2 ENC | 5 |
| 1.3 DDA & ENC | 6 |
| 1.4 DAC | 7 |
| 1.5 ADC | 8 |
| 1.6 PCL | 9 |
| 2. PULSE OUTPUT TRIGGERED BY DDA CYCLE INTERRUPT | 10 |
| 3. DDA PULSE OUTPUT CONTROL AND THE USE OF CYCLE INTERRUPT | 11 |
| 4. ACQUIRING DDA FIFO FLAG STATUS | 12 |
| 5. DDA PULSE OUTPUT CONTROL AND THE USE OF FIFO INTERRUPT | 13 |
| 6. DDA OUTPUT PULSE COUNTER CONTROL | 14 |
| 7. EMERGENCY STOP DURING PULSE OUTPUT | 15 |
| 8. DDA PULSE OUTPUT CONTROL | 16 |
| 9. ACQUIRING THE NUMBER OF COMMANDS IN DDA FIFO..... | 17 |
| 10. DDA FIFO INTERRUPT | 18 |
| 11. ENABLING INTERRUPT TRIGGERED BY ENCODER (ENC) INDEX (Z PHASE) | 19 |
| 12. ENABLING ENCODER COMPARISON VALUE INTERRUPT | 20 |
| 13. ENABLING ENCODER INPUT | 21 |
| 14. ENABLING ENCODER LATCH | 22 |
| 15. DAC ANALOG VOLTAGE OUTPUT TRIGGERED BY ENCODER COMPARATOR..... | 23 |
| 16. TIMER INTERRUPT..... | 24 |
| 17. ADC COMPARATOR INTERRUPT CONTROL..... | 25 |

| | |
|--|-----------|
| 18. ACQUIRING ADC INPUT VOLTAGE | 26 |
| 19. ACQUIRING ADC VOLTAGE CONVERSION STATUS | 27 |
| 20. ENABLING ADC TAG CHANNEL INTERRUPT | 28 |
| 21. DAC ANALOG VOLTAGE OUTPUT TRIGGERED BY ADC INPUT VOLTAGE COMPARATOR..... | 29 |
| 22. ACQUIRING ADC INPUT VOLTAGE (SINGLE CONVERSION) | 30 |
| 23. DAC ANALOG VOLTAGE OUTPUT..... | 31 |
| 24. DAC ANALOG VOLTAGE OUTPUT TRIGGERED BY LIO (LOCAL INPUT/OUTPUT) | 32 |
| 25. RIO INPUT ACQUISITION AND INTERRUPT CONTROL | 33 |
| 26. ACQUIRING INPUT AND SETTING OUTPUT OF RIO CONTROL..... | 34 |
| 27. RIO TRANSMISSION ERROR INTERRUPT CONTROL | 35 |
| 28. ACQUIRING RIO TRANSMISSION STATUS..... | 36 |
| 29. DAC ANALOG VOLTAGE OUTPUT TRIGGERED BY RIO INPUT | 37 |
| 30. PROGRAMMING FOR PCL HARDWARE POSITION CLOSED-LOOP CONTROL AND THE USE OF DDA CYCLE INTERRUPT..... | 38 |
| 31. PCL HARDWARE POSITION CLOSED-LOOP CONTROL | 39 |

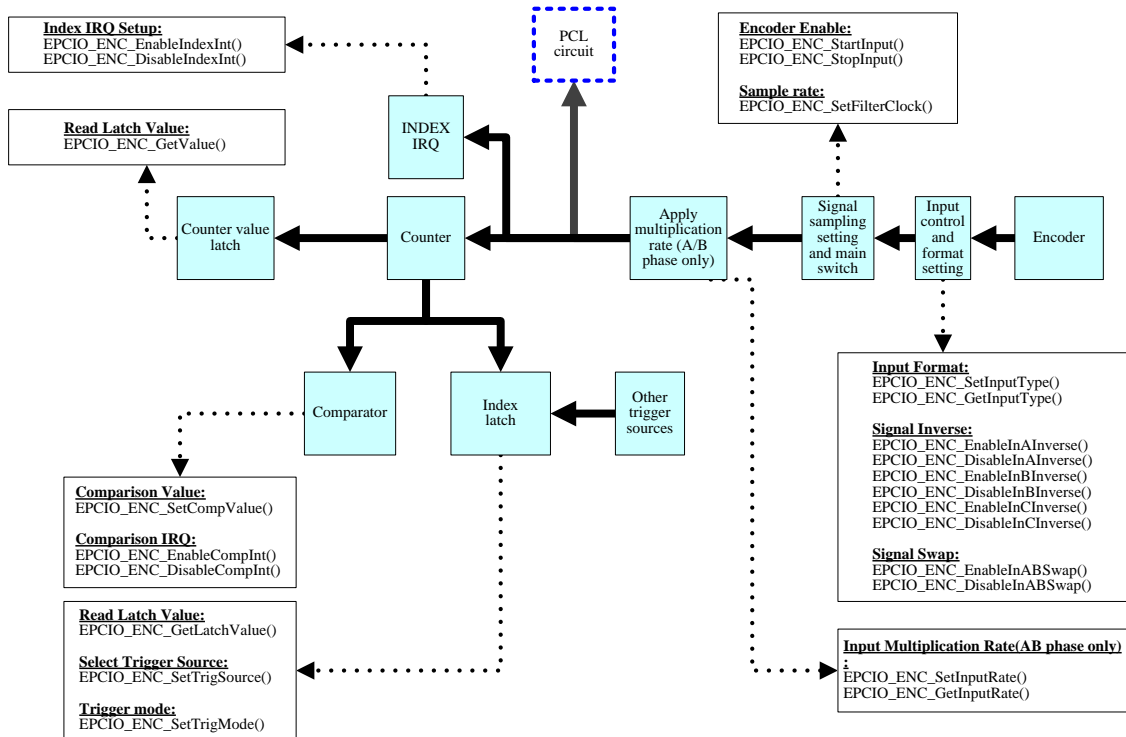
1. Device Driver Calling Flowcharts

The installation CD-ROM includes the source files of the device driver library testing tools, which source files are available for use by the user. The user can learn to use the device driver library by referring to the following flowcharts.

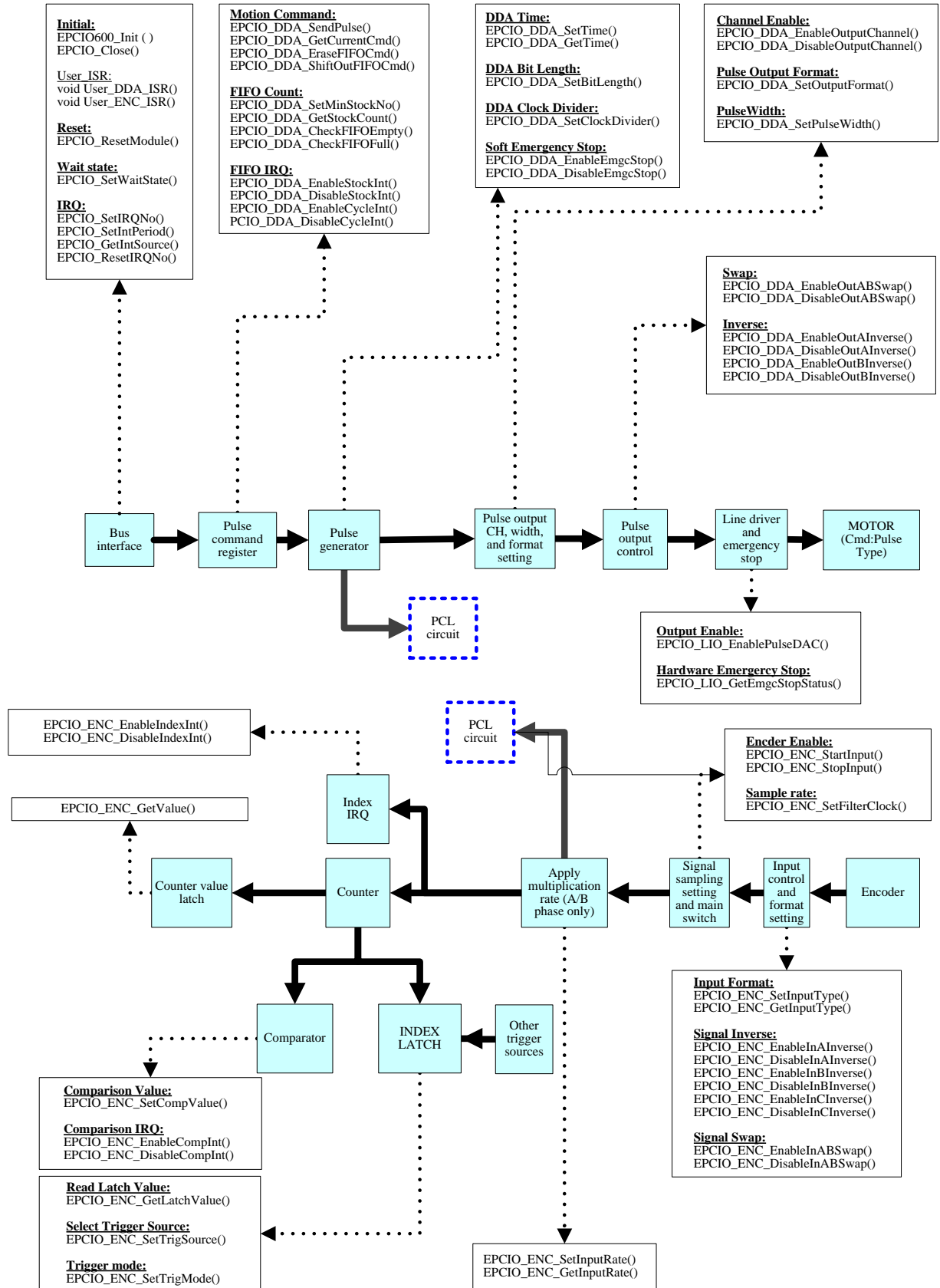
1.1 DDA



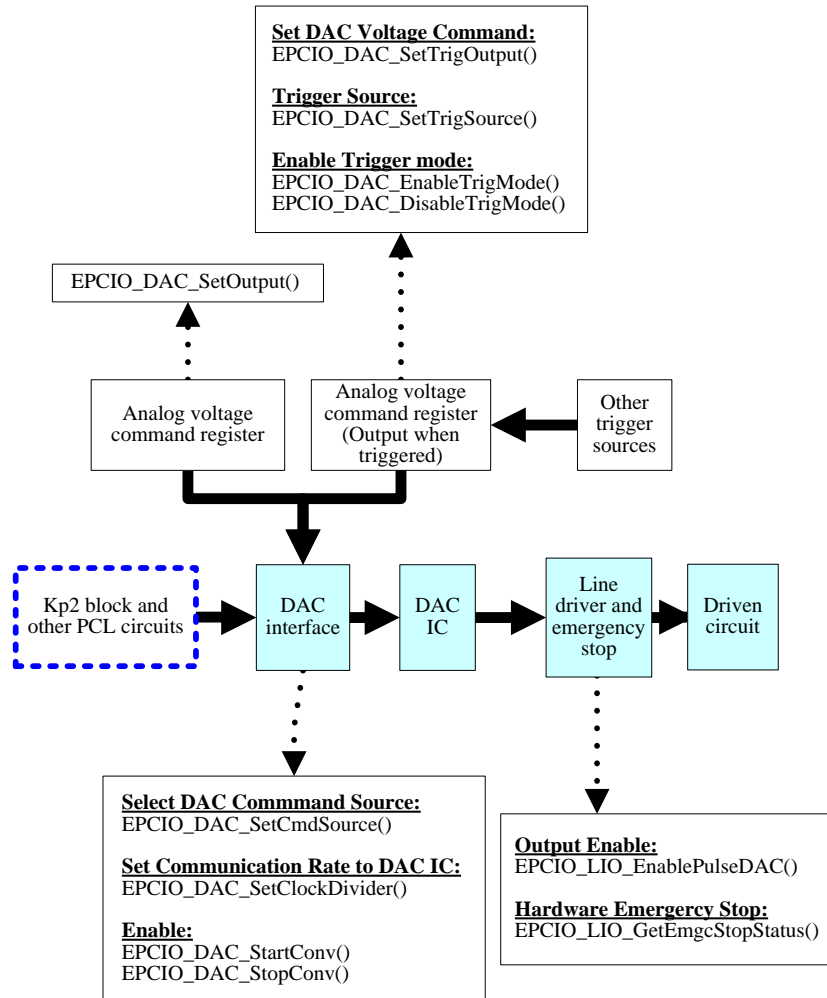
1.2 ENC



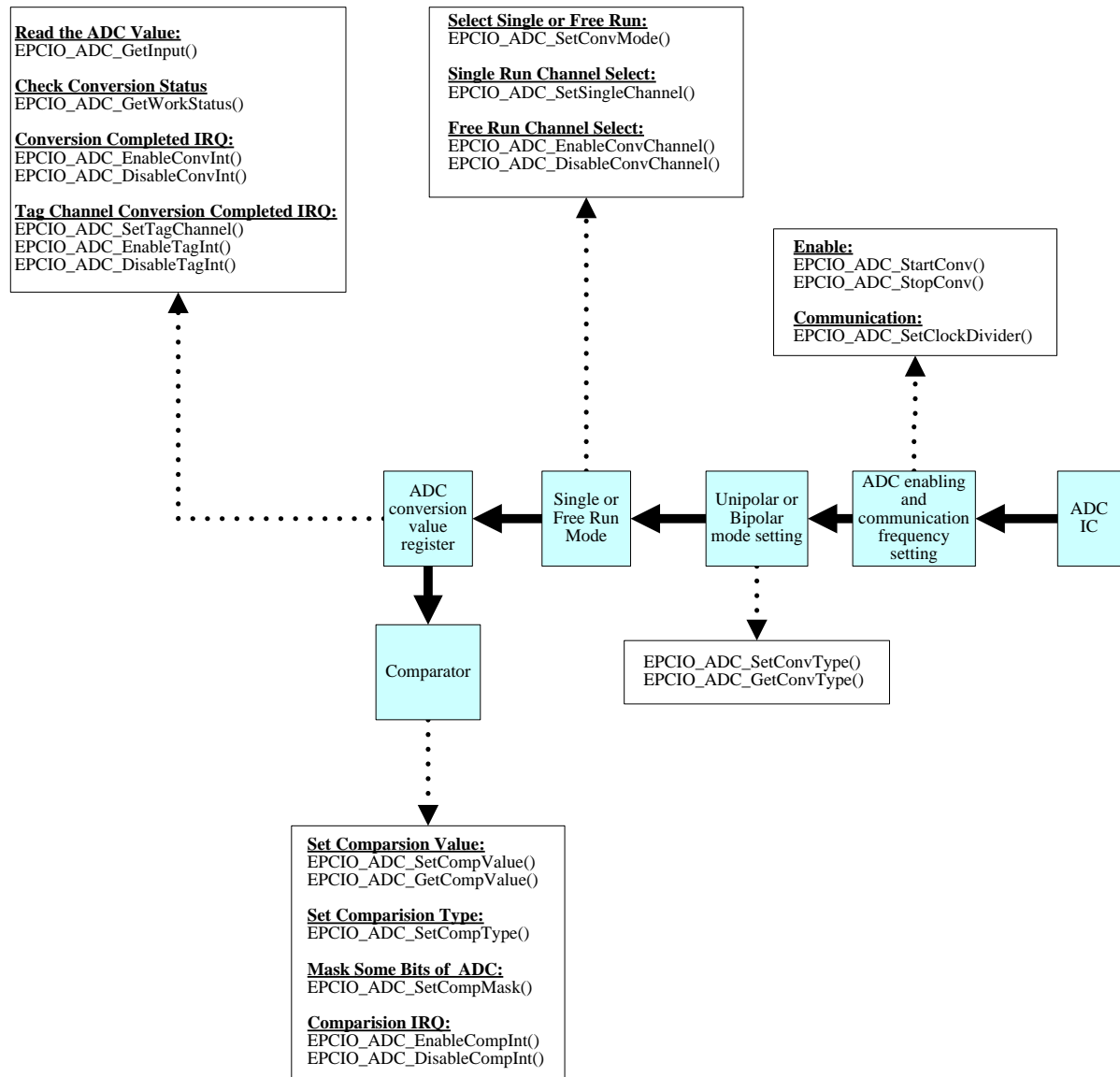
1.3 DDA & ENC



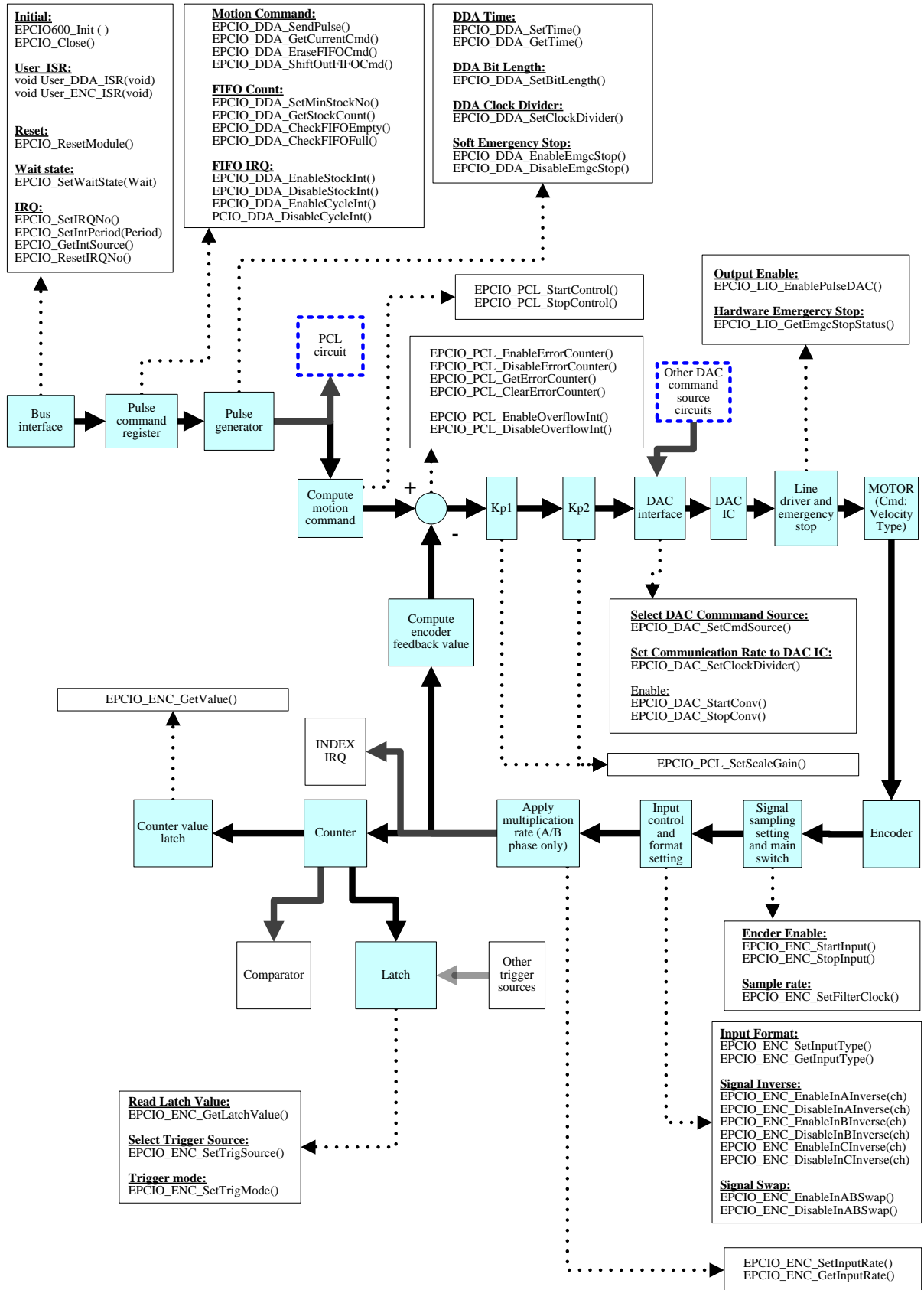
1.4 DAC



1.5 ADC



1.6 PCL



2. Pulse Output Triggered by DDA Cycle Interrupt

■ Example program

CycleInt.cpp

■ Description

This example program enables DDA cycle interrupt in order for the interrupt service routine to send a fixed number of pulses from channel 0.

3. DDA Pulse Output Control and the Use of Cycle Interrupt

■ Example program

CycPulse.cpp

■ Description

This example program sends pulses to channel 0 and programs a DDA cyclic interrupt which occurs at a fixed clock frequency in order for the interrupt service routine to call the next command.

4. Acquiring DDA FIFO Flag Status

■ Example program

FIFOChk.cpp

■ Description

This example program acquires the FIFO status of DDA channel 0.

5. DDA Pulse Output Control and the Use of FIFO

Interrupt

■ Example program

FIFOCtrl.cpp

■ Description

This example program sends pulses from channel 0 and programs a DDA FIFO interrupt. When the number of commands in the FIFO is smaller than the preset minimum number, an interrupt will be triggered so that the interrupt service routine is automatically called. The interrupt service routine will increase the number of commands in the FIFO to the safe minimum value.

6. DDA Output Pulse Counter Control

■ Example program

PulseCnt.cpp

■ Description

This example program sends pulses from channel 0 and acquires the number of pulses from the internal EPCIO pulse counter that are actually output.

7. Emergency Stop During Pulse Output

■ Example program

EmgcStop.cpp

■ Description

This example program sends pulses from channel 0 and programs the emergency stop function.



8. DDA Pulse Output Control

■ Example program

PulseOut.cpp

■ Description

This example program sends pulses from channel 0.

9. Acquiring the Number of Commands in DDA FIFO

■ Example program

StockChk.cpp

■ Description

This example program acquires the number of commands that are stored in the FIFO of DDA channel 0 but have yet to be executed.

10. DDA FIFO Interrupt

■ Example program

StockInt.cpp

■ Description

This example program enables DDA FIFO interrupt of channel 0. When the number of commands in the FIFO of channel 0 is equal to the preset minimum value (i.e., 0), an interrupt will be triggered so that the interrupt service routine is automatically called. This function is similar to DDA cycle interrupt; however, while DDA cycle interrupt occurs at a fixed frequency, DDA FIFO interrupt is triggered only when the number of commands in the FIFO meets the preset condition.

11. Enabling Interrupt Triggered by Encoder (ENC) Index (Z Phase)

■ Example program

ENCIndex.cpp

■ Description

This example program programs input into encoder channel 0 and sets the encoder index interrupt function. An interrupt will be triggered when the encoder index (Z phase) is input.

12. Enabling Encoder Comparison Value Interrupt

■ Example program

ENCComp.cpp

■ Description

This example program demonstrates input into encoder channel 0 and enables encoder comparison value interrupt. An interrupt will be triggered when the count of the encoder counter is equal to the comparison value.



13. Enabling Encoder Input

■ Example program

ENCCtrl.cpp

■ Description

This example program enables input into encoder channel 0.

14. Enabling Encoder Latch

■ Example program

ENCLatch.cpp

■ Description

This example program demonstrates the input settings of encoder channel 0 and sets the index of encoder channel 0 as the trigger source. When the index (Z phase) of encoder channel 0 is input, encoder latch will be triggered so that the value recorded in encoder channel 0 is recorded into latch register of channel 0.

15. DAC Analog Voltage Output Triggered by Encoder Comparator

■ Example program

ENCTrig.cpp

■ Description

This example program uses an encoder counter comparator to trigger the DAC so that the DAC outputs a preset voltage according to a software setting.

16. Timer Interrupt

■ Example program

Timer.cpp

■ Description

This example program uses the timer function so that the timer triggers an interrupt at a fixed time.

17. ADC Comparator Interrupt Control

■ Example program

ADCComp.cpp

■ Description

This example program sets the comparison value of ADC's channel 0 comparator. When the comparison condition is met, with the voltage passing the comparison value while decreasing, an interrupt will be triggered. This program will perform ADC conversion continuously, thereby continuously triggering interrupt when the comparison condition is met.

18. Acquiring ADC Input Voltage

■ Example program

ADCInput.cpp

■ Description

This example program uses ADC channel 0 for continuous positive and negative voltage conversion (-10 ~ 10 volt) and acquires the input voltage value.



19. Acquiring ADC Voltage Conversion Status

■ Example program

ADCState.cpp

■ Description

This example program acquires the ADC conversion status.

20. Enabling ADC Tag Channel Interrupt

■ Example program

ADCTag.cpp

■ Description

This example program sets ADC channel 0 as the tag channel. Voltage conversion completed in the tag channel will trigger an interrupt. This program will perform analog-to-digital conversion continuously, thereby continuously triggering interrupt.

21. DAC Analog Voltage Output Triggered by ADC Input Voltage Comparator

■ **Example program**

ADCTrig.cpp

■ **Description**

This example program uses an ADC comparator to trigger the DAC so that the DAC outputs a preset voltage according to a software setting.

22. Acquiring ADC Input Voltage (Single Conversion)

■ Example program

ADC1Time.cpp

■ Description

This example program uses ADC channel 0 to conduct single positive and negative voltage conversion (-10 ~ 10 volt) and to acquire the input voltage value.

23. DAC Analog Voltage Output

■ Example program

DACDirWR.cpp

■ Description

This example program demonstrates how to program DAC analog voltage output.

24. DAC Analog Voltage Output Triggered by LIO (Local Input/Output)

■ Example program

LIOTrig.cpp

■ Description

This example program demonstrates DAC analog voltage output via a software setting and uses a local input to trigger DAC analog voltage output.

25. RIO Input Acquisition and Interrupt Control

■ Example program

RIOInput.cpp

■ Description

This example program acquires the signal at the input of slave 0 in Remote I/O set 0 and uses the interrupt function of this input. When the input into slave 0 changes from LOW to HIGH, an interrupt will be triggered so that the interrupt service routine is automatically called.



26. RIO Read/Write Control

■ Example program

RIOCtrl.cpp

■ Description

This example program demonstrates acquiring input and setting output of slave 0 in Remote I/O set 0.

27. RIO Transmission Error Interrupt Control

■ Example program

RIOError.cpp

■ Description

This example program demonstrates transmission for slave 0 in Remote I/O set 0 and enables transmission error interrupt. When a transmission error occurs, an interrupt will be triggered so that the interrupt service routine is automatically called.

28. Acquiring RIO Transmission Status

■ Example program

RIOState.cpp

■ Description

This example program demonstrates transmission for slave 0 in Remote I/O set 0 and acquires the current transmission status.

29. DAC Analog Voltage Output Triggered by RIO Input

■ Example program

RIOTrig.cpp

■ Description

This example program uses DAC analog voltage output via a software setting in order for the remote input of slave 0 in Remote I/O set 0 to trigger the output.

30. Programming for PCL Hardware Position Closed-Loop Control and the Use of DDA Cycle Interrupt

■ Example program

PCL_DDA.cpp

■ Description

This example program uses position control loop (PCL) for channel 0. Commands are output via DDA cycle interrupt and sent to the targeted positions by the interrupt service routine.



31. PCL Hardware Position Closed-Loop Control

■ Example program

PCLCtrl.cpp

■ Description

This example program uses position control loop (PCL) for channel 0.