®

Control Technology Inc.
5734 Middlebrook Pike
Knoxville, TN 37921

# CTI Workbench Samples Guide

V 1.03 – May 26, 2016

The following samples are installed to the Public Documents area of your PC (typically this location is either "C:\Users\Public\Documents\Control Technology\CTIWorkbench\Samples" or "C:\Documents and Settings\All Users\Documents\Control Technology\CTIWorkbench\Samples" ).   You must change the IP address in the project settings to match your product if you wish to download and execute them on a CTI target.

| Project Name | IEC-61131 Language | Required Hardware | Description |
|---|---|---|---|
| ACP1_FlashST, ACP1_FlashLD, ACP1_FlashFBD | ST, LD, FBD | 2500P_ACP1 | Allows the user to control the USER LED on the product faceplate by manipulating user settings in the product web page. Demonstrates use of CTI_ACP1_CONFIG() and CTI_USER_LED() functions. |
| ACP1_Modbus | LD | 2500P_ACP1 | Uses a single rung of ladder to increment an integer. This integer is made available for reads and/or writes via Modbus TCP Slave/Server.  A Modbus TCP Master/Client connection is made to this server and will perform slower periodic read requests of this changing value. NOTE: Before using this sample, the Modbus Client IP Address must be changed to reflect that of the ACP1 |
| ACP1_Serial | ST | 2500P_ACP1 | Demonstrates how to use the serial port. When connected to a PC terminal via null modem, the user is prompted every 10 seconds to enter a number of flashes.  The number is sent back to the PC, and the USER LED flashes the specified number of times. NOTE: Please have your terminal serial port set to 9600,8,N,1. |
| DataLog | FBD, ST (sub-program) | 2500P_ACP1, 2500-CXXX, FTP Server (PC) | This sample can demonstrate data logging in Simulation mode, but it must run on CTI Target to show all features including data logging to SD Card and FTP file transfer. FTP Transfer requires setup of FTP Server on a network-accessible PC. After downloading project to ACP1, set application to RUN mode. Data logging starts when BOOL variable 'Logging' is forced to TRUE. Operation of the 'CTI_DataLog' sub-program is described by comments within the program. You can use this sub-program "as-is" in any other project simply by copying it to that project. Also, it can be used as a template and modified to meet specific needs of the application. NOTE:  If ACP1 is used with CTI 2500 Series CPU, the ACP1 RTC can be automatically synchronized to the PLC clock via CTI Data Cache interface. The PLC IP Address must be updated in the 'Fieldbus Configuration' screen. Additionally, the FTP Server settings must be configured for ACP1 module.  Both PLC time synchronization and FTP Server are configured in the ACP1 web server 'Module Configuration' page. |
| ACP1_CacheIO | FBD | 2500P_ACP1, 2500-CXXX | Uses a single function block to increment an integer. This integer's value is delivered to variable V1 in the 2500-CXXX PLC through the data cache fieldbus interface (CTI 2500 Data Cache).  The CAMP Client fieldbus protocol is then used to periodically read V1 back from the 2500-CXXX PLC in to a separate ACP1 variable for verification. NOTE: Before using this sample, the CAMP Client IP Address must be changed to reflect that of the PLC, and the PLC IP Address must be updated in the CTI 2500 Data Cache configuration to reflect that of the PLC. |
| ACP1_NormalIO | ST | 2500P_ACP1, 2500-CXXX or 505 PLC | Uses a single statement to increment an integer. This integer is transferred to the 2500-CXXX or compatible 505 PLC via the Normal I/O interface ( CTI 2500P-ACP1 I/O ).  If the ACP1 is properly configured within the PLC (i.e. 32/32/32/32 at I/O Address 1), the user can monitor WX65 to see this counter within the PLC. Additionally WY97 can be set within the PLC, and this value will be transmitted to a separate ACP1 variable. NOTE: Before using this sample, the PLC I/O configuration must include the I/O addresses assigned to the ACP1, and the PLC and ACP1 I/O configurations must match. |

| | | | |
|---|---|---|---|
| **EIP_Adapter_1** | ST | 2500P_ACP1, EIP I/O Scanner | Provides an example for ACP1 serving as an Ethernet/IP I/O Adapter performing an I/O data transfer with an Ethernet/IP I/O Scanner. NOTE: To make the demo execute correctly, an EIP I/O Scanner must be configured and connected to the ACP1. In most cases, a Rockwell PLC or a second ACP1 module will serve as the Ethernet/IP I/O Scanner. |
| **EIP_Scanner_1** | ST | 2500P_ACP1, EIP I/O Adapter | Provides an example for ACP1 serving as an Ethernet/IP I/O Scanner that establishes a connection to an Ethernet/IP I/O Adapter and performs I/O data transfer. NOTE: To make the demo execute correctly, an EIP I/O Adapter be configured and connected to the ACP1. Various Ethernet/IP compatible devices such as bar code scanners, drive controllers, I/O controllers, or a second ACP1 module can serve as the EIP I/O Adapter. |
| **EIP_TagClient_1** | ST | 2500P_ACP1, EIP Server | Provides an example for ACP1 serving as an Ethernet/IP Tag Client that establishes a connection to an Ethernet/IP Server and transfers data via client/server tag names. NOTE: To make the demo execute correctly, a Ethernet/IP Tag Server be configured and connected to the ACP1. In most cases, a Rockwell PLC will serve as the Ethernet/IP Server. |
| **TCP_UDP_Client** | ST | 2500P_ACP1, Workbench simulator | Provides a working demo and template for developing applications for the ACP1 to be a TCP (and/or UDP) Client that sends/receives Ethernet packets containing user-specified protocol data. Two different programs are included – one each for TCP and UDP messaging services. The two programs can run independently or concurrently. NOTE: This sample can run with only one ACP1 module and Workbench. Workbench can act as TCP (and/or UDP) Server (by running 'TCP_UDP_Server' project in simulation mode) after this project is configured (Server IP address set) and downloaded to the ACP1. |
| **TCP_UDP_Server** | ST | 2500P_ACP1, Workbench simulator | Provides a working demo and template for developing applications for the ACP1 to be a TCP (and/or UDP) Server that receives/sends Ethernet packets containing user-specified protocol data that are generated from a remote TCP (and/or UDP) Client. Two different programs are included – one each for TCP and UDP messaging services. The two programs can run independently or concurrently. NOTE: This sample can run with only one ACP1 module and Workbench. Workbench can act as TCP (and/or UDP) Client (by running 'TCP_UDP_Client' project in simulation mode) after this project is downloaded to the ACP1.The ACP1 IP address must be set in the 'TCP_UDP_Client' project before it is started in simulation mode. |
| **CoffeeMachine** | SFC and ST | None | This sample can work in simulation mode or on a CTI Target. A vending machine is drawn on the PC using Workbench Graphics with which the user can interact by clicking buttons. NOTE: When running in Workbench, please click on Graphics->Machine located in the workspace area to view and interact with the vending machine |
| **FBDDemo** | FBD | None | This sample can work in simulation mode or on a CTI Target. It demonstrates how a flashing output can be achieved. To interact with it, one must force the input values to change by clicking on them. |
| **Graphics** | ST | None | This sample can work in simulation mode or on a CTI Target. A holding tank is drawn on the PC using Workbench Graphics with which the user can interact by clicking buttons. NOTE: When running in Workbench, please click on Graphics->MainGraphic located in the workspace area to view and interact with the tank. |
| **LightGame** | ST | None | This sample can work in simulation mode or on a CTI Target. A game is drawn on the PC using Workbench Graphics with which the user can interact by clicking buttons. NOTE: When running in Workbench, please click on Graphic->Main located in the workspace area to view and interact with the game. |
| **SFCDemo** | SFC | None | This sample can work in simulation mode or on a CTI Target. If the variable BRun is forced to true, this sample demonstrates cycling between 4 separate outputs. |