UTC Series

UtcPanel
QUICK SETUP GUIDE

Micro Trend Automation Co., Ltd
3F, No.78, Cheng Kung Road, Sec. 1
Nan Kang, Taipei, Taiwan
TEL:(02)27882162   FAX:(02)27857173
Contents

Chapter 1. Service Level Guide.......................................................................................... 02
  1.1 Setup UtcPanel_Eng On A Computer ........................................................................ 02
    1.1.1 Download UTC System Setup Software From Micro Trend Website .................. 02
    1.1.2 Install the Application Software UtcPanelEng On the Computer ................. 05
    1.1.3 Connecting RS232 Cable From the Computer to UTC Controller ................. 07
    1.1.4 Run the UtcPanelEng On the Computer ...................................................... 08
    1.1.5 Communication Failure between Computer and UTC ................................... 09
  1.2 Using UtcPanelEng To Do the UTC Control Setup or Maintenance .................... 11
    1.2.0 Get some information from the controller through “Online Commands” window .......................................................................................................................... 11
    1.2.1 Download the UTC Default Setup File ....................................................... 14
    1.2.2 Download the User Program File or Backup File ....................................... 15
    1.2.3 To Activate the Control System .................................................................. 16
    1.2.4 Saving All the Control Programs And Variables In the Flash ROM .......... 17
    1.2.5 Saving All the Control Programs And Variables In a File to the Computer ................................................................................................................................. 18
    1.2.6 Restore the Data saved in Flash ROM to the SRAM ................................. 19
    1.2.7 UTC Firmware Update .............................................................................. 20
    1.2.8 Some More Online Commands On Service Propose ................................ 22
    1.2.9 Some Parameters Regarding Control Setup .............................................. 23
Chapter 2. Designing With UTC Control........................................................................ 31
  2.1 A cut to length with timer cut tool output control.................................................. 31
    2.1.1 The details of project requirement .............................................................. 31
      2.1.1.1 Parameter Settings ........................................................................... 31
      2.1.1.2 Control System Consideration ......................................................... 32
    2.1.2 Start the Design ....................................................................................... 32
      2.1.2.1 Hardware Assignments .................................................................... 32
      2.1.2.2 Internal Variable Assignments ........................................................ 32
      2.1.2.3 PLC programs planning example ..................................................... 33
      2.1.2.4 PLC and motion programs coding example ..................................... 33
  2.2 A cut to length with two sensors cut tool output control, mark signal capture and Hitech ADP3 as HMI ......................................................................................... 34
    2.2.1 The details of project requirement .............................................................. 34
      2.2.1.1 Parameter Settings ........................................................................... 34
      2.2.1.2 Control System Consideration ......................................................... 35
    2.2.2 Start the Design ....................................................................................... 35
      2.2.2.1 Hardware Assignments .................................................................... 35
      2.2.2.2 Internal Variable Assignments ........................................................ 35
      2.2.2.3 PLC programs planning example ..................................................... 36
  2.3 A single axis motion program example for drilling cycle and Hitech ADP3 as HMI ............................................................................................................................. 37
    2.3.1 The details of project requirement .............................................................. 37
    2.3.2 Start the Design ....................................................................................... 37
Chapter 3. Appendix...................................................................................................... 38
  3.1 PROJECT_1 ......................................................................................................... 38
  3.2 PROJECT_2 ......................................................................................................... 48
Chapter 1. Service Level Guide

This is a step-by-step quick reference for an uninitiated user to setup a UTC system. It is essential to have a computer with RS232 communication port for UTC system setup. For some latest model notebook computers, which have a few USB ports but no RS232 port, please purchase a USB-to-RS232 adaptor and setup the adaptor prior to start the UTC system setup. As well, a panel software running with Windows Operating System, Windows 95 or higher version, is needed for the setup. Please contact Micro Trend to get an application disk or visit Micro Trend website to download the panel software.

1.1 Setup UtcPanel_Eng On A Computer

In this section, we will show the novice how to connect a computer to the UTC control. If you have already successfully connected please skip this section.

1.1.1 Download UTC System Setup Software From Micro Trend Website.

Please visit: http://www.utrend.com.tw and click “SERVICES” from any page of the site. Then enter name: utrend and password: 7173. In the service page, two different versions of the UTC operation manual and the setup software will be found. Please download the execution software – UtcPanel_Eng. from this page then run it on the computer. After running this execution file, we get a directory named UtcPanelEng with two sub-directories named Disk1 and Disk 2.
a. From Micro Trend Website, click “SERVICES”

![Fig1.1](image1)

b. Then enter name: utrend and password: 7173

![Fig1.2](image2)

c. Download the execution software – UTCPanel_Eng

![Fig1.3](image3)
d. Save UTC-SetupEng.exe

Fig1.4
1.1.2 Install the Application Software UtcPanelEng On the Computer

From Micro Trend application disk or downloaded resource, we get a directory named Utc-SetupEng with two sub-directories named Disk1 and Disk2. In the Disk1 sub-directory, there is a setup execution file. Please run this setup file to install the application software named Utc-SetupEng. During the installation process, please fill all three columns of user name, company and serial number. The serial number can be any number. When the installation process finished, an icon named Utc-SetupEng will be shown on the desktop screen. Please connect the RS232 cable and turn the UTC control power on before clicking the UtcPanelEng icon as Fig1.1.

The icon.

The icon.

Fig1.5

Fig1.6

a. In the Disk1 sub-directory, there is a SETUP execution file.
b. please fill all three columns of user name, company and serial number. The serial number can be any number.
1.1.3 Connecting RS232 Cable From the Computer to UTC Controller

The connection cable for computer to UTC controller is a one-to-one 9 pin D-connector with pin 9 cut off. Please make the connection when control power is turned off. The default protocol for UTC COM1 is 38400, N, 8, 1 with RTS/CTS flow control. It can be changed by setting the dipswitch 2 and 3 on the control board. The default protocol for UTC COM2 is 19200, N, 8, 1 without flow control. It can be changed by I-Parameter 2 and 3 settings.
1.1.4 Run the UtcPanelEng On the Computer

With the RS232 cable connected, computer serial COM port properly installed and control power turned on, click the UtcPanelEng icon on the desktop. The information “Can not find control card!” will be shown on the screen.

a. The information “Can not find control card!” will be shown on the screen.

Please click “Setup” button, under the message, to setup the RS232 connection protocol, including COM port number, BAUD rate, delay time and flow control handshake (CTS/RTS). Please correctly set the computer COM port number, BAUD rate 38400, delay time 50000 and select CTS/RTS to activate flow control. Then click “OK” to approach again communicating with the UTC control. The information “Card found on Comx” will be shown on the screen to indicate that the software setup process has been finished and the communication between computer and UTC control is successful. If still “No control card found”, please follow section 1.1.5 to do the diagnostic process.

b. The information “Card found on Comx” the communication between computer and UTC control is successful

If this screen shows up, it indicates that the connection is successful; Click “OK” to enter the panel page
1.1.5 Communication Failure between Computer and UTC

The communication failure problem could be at computer side, controller side or the connection. Please re-check the one-to-one connection cable, pin 9 should not be connected, and verify the computer serial port settings. At UTC controller side, please take the following steps to identify the problem.

a. If the message shows “Can’t open communication port!”
   
   It indicates that the problem is at the computer communication port; either set to a non-exist port number or this port has been occupied by other application software. Please check the computer property and then set it to right port number or turn off other application software that occupies this communication port.

   **Sometimes this problem could happen by double clicking the icon twice to cause the second approach failure.**

b. Check the dipswitch 2 and 3 settings.
   
   Please turn both the dipswitch 2 and 3 off if any of them not been turned off. Then re-turn-on the control power and try again to do the connection. The dipswitch 2 and 3 decide the COM1 BAUD rate. When both of them turned off, the COM1 BAUD rate is 38400. Please use the following table for other BAUD rate settings.
c. Check the “W. Dog” LED. This LED should turn on for a few seconds then turn off after power up. When “W. Dog” LED turns on, the control does not do any communication or run any user programs; it indicates that the control is in an abnormal situation.

d. What to do if the “W. Dog” LED does not turn off after power up? Please turn the dipswitch 1, on the control board, to on position then power up. This action reset the all the UTC parameters to the default values and clear all the preloaded motion programs and PLC programs. If the LED still keeps turned on all the time after this process, the control board or the firmware chip should be replaced. Please contact Micro Trend to request for service.

e. Try using COM2 instead of COM1. If the “W. Dog” LED turns off after the reset process but still unable to communicate with COM1, please turn off the control power and switch the connection to COM2. Then try COM2 with BAUD rate 19200 and no CTS/RTS flow control.

f. If the computer still cannot find the UTC control after all these approaches, please contact local service people or Micro Trend for help.

<table>
<thead>
<tr>
<th>NO.2</th>
<th>NO.3</th>
<th>COM1 baud rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>38400</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>19200</td>
</tr>
<tr>
<td>OFF</td>
<td>ON</td>
<td>9600</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>115200</td>
</tr>
</tbody>
</table>
1.2 Using UtcPanelEng To Do the UTC Control Setup or Maintenance

In this section, we will explain the setup and program update procedure using a computer running UtcPanelEng software. These procedures are normally for a service engineer who would like to setup or update software to a new machine with a fully tested user program.

1.2.0 Get some information from the controller through “Online Commands” window.

Once the computer successfully finds the control card by running “UtcPanelEng”, we can start the service level actions from the computer. We can download the basic setup file for different versions, change parameter or variable settings, monitor the system status, download motion programs or PLC programs, issue the Online-Commands, backup system configurations or update firmware. Now, we first get some information from the controller through “Online Commands” window. Also, we will request the controller to take some actions upon our Online Commands.

a. Click “START” to get the Online Commands window. Fig 1.2.0.1
b. Click “Tools” the “Online Commands” shown in Fig 1.2.0.2
As shown in the Fig 1.2.0.3, we can see that the control reports information or takes action upon the online commands we issue from the window. Actually, the control report to any serial port, which sent commands to request information. Also, it takes action upon receiving any online command from any serial port even from a PLC program, which is enabled. A full set of “Online Commands” explanation can be found in the UTC control manual “Online Commands” section.
1.2.1 Download the UTC Default Setup File.

For each different model or version of UTC, Micro Trend preloads in the controller before shipment a dedicated default setup file bases on customers’ orders. This basic setup file is a set of default M-Variable-definitions. These M variables are generally used in most of the PLC programs or motion programs. If the control firmware version has been changed or a hardware or software reset process has been performed, we should re-download the basic setup file. The following table shows the files for various models and versions.

<table>
<thead>
<tr>
<th>Model</th>
<th>Version</th>
<th>M-Definition File</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTC400P/200P/100P</td>
<td>2.2x/2.1x</td>
<td>400P220.UTC</td>
</tr>
<tr>
<td>UTC400V/200V/100V</td>
<td>2.2x/2.1x</td>
<td>400V220.UTC</td>
</tr>
</tbody>
</table>

However, each of these M-Variable-Definitions can be redirected to different memory location for different purposes.

Please follow these procedures to download the default setup file to the controller.

a. For first time setup, before the downloading, please use the online command “$$$$***” to clear all the memory in the controller.

b. Click “File” the “Download to UTC” then “File” as shown in Fig 1.2.1.1

![Fig 1.2.1.0](image1.png)

![Fig 1.2.1.1](image2.png)
c. Select the right file type and then a target default setup file from the computer and then click “Open”

![Fig 1.2.1.2](image)

![Fig 1.2.1.3](image)

d. It takes a few seconds to minutes for the downloading depends on the file size. There is a red bar to show the percentage of completed portion during download process, as Fig 1.2.1.3.

1.2.2 Download the User Program File or Backup File

To activate a UTC control system, there must be some PLC programs and/or some motion programs pre-loaded in the controller and activated. We call them user application programs or user programs. Normally, the user programs are put in a file; we call it user program file; with extension UTC. After a control system is fully tested, we may upload all the parameter and variable settings as well as all the user programs then store these items in a file. We call it backup file, with extension DAT. Please be aware that these UTC or DAT files are with simple ASCII text format. The process is the same as downloading the default setup file only choosing different file. Normally, the user program file should be loaded after the setup file. Otherwise, it is possible that some parameters, which have been set to match an individual application, will be reset to the default value and cause mal-function.
1.2.3 To Activate the Control System

After we go through the process 1.2.0 to 1.2.2, we have a well-tested user program loaded in the UTC controller. We may activate the control system at this moment if the initial variables and parameters are included in the user program. Otherwise, we might need to set some parameters or variables for an individual system before activating the control. For example, the unit ratio between the encoder counts and the customer define unit. It could be various for each individual machine and leave for the machine setup engineer to finish the settings.

There are some possible ways to activate the system control after first time the software is loaded in the UTC. It is flexible to allow the users to choose their own way to activate the control system. The following is some typical ways to activate a UTC control system.

1. Activates by setting PLC control parameter I6.
   - Set the I6 to 1, 2 or 3 to activate the PLC control programs.
   - In most situations, we write PLC programs to control the whole system, including starting a motion program. Set I6 to a suitable value can activate all the system PLC programs immediately.

2. Activate an individual PLC program by the PLC enable command.
   - We may use the command “enaplc x” (x=0 to 15) to test an individual PLC program.

3. To run a motion program by the motion program enable command.
   - We may use the command “BxR” (x= program number) to start a motion program running.
1.2.4 Saving All the Control Programs And Variables In the Flash ROM.

It is very important for a working control system to save all the content in the working space, SRAM with backup battery, to the Flash ROM space. In case someday the data in the SRAM are destroyed by low battery or any other reason, we can restore all the content back to the SRAM from the Flash ROM.

The saving process should be done whenever:
- The machine setup is first time finished.
- The PLC programs or motion programs are modified.
- The compensation data is changed.
- The EEPROM (Flash ROM) is replaced.
- The UTC system firmware is updated
- Any other variables settings are changed

1. Before processing the saving process, please type from the online commands window to disable currently running motors and PLC programs.
   >>>^A ;Abort currently running motion program.
   >>>^D ;Disable currently running PLC program.
   >>>^K ;Disable currently enabled motors

2. To do the saving process, we simply type “Save” in the online commands window.
   >>>Save ;To start the saving process
3. The saving process will take a few seconds; please do not turn the control power off during the saving process.

4. After the saving process, we may re-turn-on the control to get the controller back to normal operations.

1.2.5 Saving All the Control Programs And Variables In a File to the Computer.

Besides saving the control data in the Flash ROM, we can also save a copy of all the control variable settings and user programs in a computer. For the next identical UTC control setup, we may download this backup file instead of the default setup file and the user program files. Please see section 1.2.2 for the download process.

The backup process is as follows.
1. Select “Backup Configurations” from the tools menu or directly from the tool bar.

![Fig 1.2.5.1](image)

2. Give a file name for this backup file and save it in a selected directory.

   The default extension of the backup file is XXXX.dat.

![Fig 1.2.5.2](image)
4. The backup process could take a few seconds to minutes. There is a red bar to show each data group processing. (Fig 1.2.5.3)

![Fig 1.2.5.3]

1.2.6 Restore the Data saved in Flash ROM to the SRAM

No matter what reason causes the control working space data losses, we may restore all the data back from the Flash ROM. There are two ways to get the data back from the Flash ROM.

1. If the RS232 communications still works fine when the control data losses happens, we may simply type “$$${}$$” in the online commands window to get the data back from the Flash ROM.

2. If the RS232 communication no longer works, please turn the control power off. And then, turn the control power on with the dipswitch 7 ON. The dipswitch location is described in section 1.1.5.

>>$$${}$$ or power up with dip switch 7 turns ON

![Static RAM vs Flash ROM]

19
1.2.7 UTC Firmware Update

The UTC control system software handles all the control behaviors including decoding the PLC and motion program, communicating with HMI through RS232, servo controls and other protection features. Micro Trend keeps updating this software to provide better features or improve the control performances. To update the UTC control system software for the hardware in the customer site, we can also use the UtcPanelEng setup software.

This is the procedure to do the system software update.

The same as the saving process, before doing the firmware upgrade, please type from the online commands window to disable currently running motors and PLC programs.

\[ \text{>>}^\text{A} \quad ;\text{Abort currently running motion program.} \]
\[ \text{>>}^\text{D} \quad ;\text{Disable currently running PLC program.} \]
\[ \text{>>}^\text{K} \quad ;\text{Disable currently enabled motors} \]

1. From the tool bar, click “Tools” then select “Firmware Upgrade”

![Fig 1.2.7.1](image1.png)

2. Select the target BIN file that stored in the computer

![Fig 1.2.7.2](image2.png)
3. The firmware upgrades process. All over the new firmware downloading process, please do not turn off the control power. Otherwise, the system firmware will be permanently destroyed. And, the control will no longer work unless we replace the Flash ROM.

   a. The flash erases process; the erasing message shows on the screen for a few seconds. Then, turn to process b to write the new bin code to the flash ROM.

   b. Write the new bin code to the flash ROM; there will be a red bar to show the downloading progress on the screen.

   ![Image of flash erasing process]

   c. After the Flash Writing completed, there will be following message shows on the screen.

   ![Image of download complete message]

   d. Now, we can turn off and re-turn-on the control power to run the new system software.

4. If the major firmware version code is changed, it indicates that there are major changes have been made in the new version. We may need to re-setup the system. In this case, please turn the control power on with dipswitch 1 in the “ON” position to re-initiate the control. And then, follow section 1.2.1 to 1.2.5 to re-setup the control.

   Please don’t forget to turn the dipswitch 1 back to off position each time after we turn it on at power up. Otherwise, the data in the SRAM area will be cleared at next power up. In that case, we will have to process 1.2.1 to 1.2.5 one more time.
1.2.8 Some More Online Commands On Service Propose

There are many online commands that are dedicated for service proposes. Please read the “UTC Motion Controller Manual” for the detail. The following are some of the examples.

>>>Type --- The control will report a short description of this firmware version

>>>Ver ---

>>>Date

>>
1.2.9 Some Parameters Regarding Control Setup

1. Online Command Summary

   Notes
   spaces     : Spaces are not important unless special noted
   {   }         : Item in {   } can be replaced by anything fitting definition
   [   ]         : Item in [   ] is an option
   [{item}…]   : Repeated syntax
   [..{item}]   : The periods are used to specify a range

   Definitions
   constant     : Number that is non-changeable
   variable     : Variable like I , M , P , Q
   expression   : Combination of constant, variable, function and operator
   data         : Constant without parentheses or expression with parentheses
   axis         : Element of coordinate system. It can be X, Y, Z, U, V, W, A, B, C

2. Global Online Command

   <CONTROL-A>

   Function   Abort all Program Execution and motor moves
   Syntax     ASCII Value 1

   <CONTROL-D>

   Function   Disable all PLC programs
   Syntax     ASCII Value 4

   <CONTROL-K>

   Function   Disable all drivers
   Syntax     ASCII Value 11
<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card Flash Reload</td>
<td>$$$$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card reset and re-initialization</td>
<td>$$$$***</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear currently opened buffer</td>
<td>CLEAR</td>
</tr>
<tr>
<td></td>
<td>CLEARALL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close the currently opened program buffer</td>
<td>CLOSE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report the controller firmware date.</td>
<td>DATE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the specified PLC(s)</td>
<td>DISPLC{constant}[,{constant}. . .]</td>
</tr>
<tr>
<td>{constant} is a positive integer representing PLC number, ranging 0~15</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable the specified PLC program.</td>
<td>ENAPLC{constant}[,{constant}. . .]</td>
</tr>
<tr>
<td>{constant} is a positive integer representing PLC number, ranging 0~15</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Syntax</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>Motor home searching</td>
<td>HM</td>
</tr>
</tbody>
</table>

**HMZ**

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do a Zero-Move Homing</td>
<td>HMZ</td>
</tr>
</tbody>
</table>

**I{constant}**

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report the specified I-variable value</td>
<td>I{constant}[..{constant}]</td>
</tr>
</tbody>
</table>

{constant} is an integer representing I variable number, ranging 0~1023

**I{constant}={expression}[:]**

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign The specified I-variable value.</td>
<td>I{constant}[..{constant}]=expression</td>
</tr>
</tbody>
</table>

{constant} is an integer representing I variable number, ranging 0~1023
The second {constant} should be greater than the first one; it represents the ending I-variable number.
{expression} is an expression
[:] means to report the value of this I-variable value.

**J+**

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jog to positive direction.</td>
<td>J+</td>
</tr>
</tbody>
</table>

**J-**

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jog to negative direction.</td>
<td>J-</td>
</tr>
</tbody>
</table>
**J/**

**Function**  Jog stop.

**Syntax**  J/

---

**J:**

**Function**  Jog a variable specified distance

**Syntax**  J:

---

**J:{constant}**

**Function**  Jog a distance specified by {constant}

**Syntax**  J:{constant}

{constant} is a floating point value representing the distance to jog in count.

---

**J=**

**Function**  Jog to a variable specified absolute position.

**Syntax**  J=

---

**J={constant}**

**Function**  Jog a motor to a specified position

**Syntax**  J={constant}

{constant} is a floating point value representing the absolute position to jog in count.

---

**J***

**Function**  Jog the addressed motor back to the motion program stop

**Syntax**  J*
**LIST PLC**

**Function** Report the existed PLC program number or specified PLC Program contents  
**Syntax** LIST PLC {constant}  
{constant} is positive integer representing the number of the PLC Program to reported, ranging 0~15.

**LISTPROG**

**Function** Report the existed motion program numbers or specified motion program contents  
**Syntax** LIST PROG{constant}  
{constant} is positive integer representing the number of the motion program to be reported.

**M{constant}**

**Function** Report the specified M variable value  
**Syntax** M{constant}[..{constant}]  
{constant} is an integer representing the variable number to be specified or the starting variable number, ranging 0 ~ 1023.  
The 2^{nd} {constant} representing the end number of variable to be reported, its value should be larger than the 1^{st} one.

**M{constant}={expression}[:]**

**Function** Set the specified M variable value  
**Syntax** M{constant}[..{constant}]= {expression}  
{constant} is an integer representing the variable number to be specified or the starting variable number, ranging 0 ~ 1023.  
The 2^{nd} {constant} representing the end number of variable to be set, its value should be larger than the 1^{st} one.  
{expression} Representing the value to be set.  
[;] means to report the value of this M variable.
M{constant}->

Function Report M variable definition
Syntax M{constant}[..{constant}]->
   {constant} is an integer representing the variable number to be
   specified or the starting variable number, ranging 0 ~ 1023.
   The 2nd {constant} representing the end number of variable to be reported,
   its value should be larger than the 1st one.

M{constant}->{*}

Function Assign M{const} as a normal variable.
Syntax M{constant}[..{constant}]->*
   {constant} is an integer representing the variable number to be specified or
   the starting variable number, ranging 0 ~ 1023.
   The 2nd {constant} representing the end number of variable to be assigned,
   its value should be larger than the 1st one.

O{constant} For Pulse Command (UTCx00P)

Function Set the output pulse frequency in the unit Pulses/msec
Syntax O{constant}
   {constant} is an integer representing the pulse number per msec, ranging
   -250~250.

OUT{constant} For Voltage Command (UTCx00V)

Function Set the output voltage
Syntax OUT{constant}
   {constant} is an integer representing the ratio of \textbf{lx30} voltage, ranging
   -100~100.
**P{constant}**

**Function**  
Report specified P variable value

**Syntax**  
P{constant}[..{constant}]

{constant} is a positive integer representing the P variable, ranging 0∼1023  
The 2\(^{nd}\) {constant} representing the end number of variable to be assigned,  
its value should be larger than the 1\(^{st}\) one.

**P{constant}={expression}[:]**

**Function**  
Set the specified P variable value

**Syntax**  
P{constant}[..{constant}]={expression}

{constant} is a positive integer representing the P variable, ranging 0∼1023  
The 2\(^{nd}\) {constant} representing the end number of variable to be assigned,  
its value should be larger than the 1\(^{st}\) one.

{expression} is the value to be assigned.

[:] means to report the value of this P variable.

**Q{constant}**

**Function**  
Report specified Q variable value

**Syntax**  
Q{constant}[..{constant}]

{constant} is a positive integer representing the Q variable, ranging 0∼1023  
The 2\(^{nd}\) {constant} representing the end number of variable to be assigned,  
its value should be larger than the 1\(^{st}\) one.

**Q{constant}={expression}[:]**

**Function**  
Set the specified Q variable value

**Syntax**  
Q{constant}[..{constant}]={expression}

{constant} is a positive integer representing the Q variable, ranging 0∼1023  
The 2\(^{nd}\) {constant} representing the end number of variable to be assigned,  
its value should be larger than the 1\(^{st}\) one.

{expression} is the value to be assigned.

[:] means to report the value of this Q variable.
<table>
<thead>
<tr>
<th>Function</th>
<th>Report the contents of the addressed buffer memory.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>RD(D(dec)/H(hex) /L(float){address}[,{length}]</td>
</tr>
<tr>
<td></td>
<td>{address} points to the buffer memory you want to list. It should be 16 bit, ranging from $0~FFFF$</td>
</tr>
<tr>
<td></td>
<td>{length} means the length of the buffer memory you want to list.</td>
</tr>
</tbody>
</table>

**SAVE**

<table>
<thead>
<tr>
<th>Function</th>
<th>Save all the contents from static RAM to FLASH ROM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>SAVE</td>
</tr>
</tbody>
</table>

**SIZE**

<table>
<thead>
<tr>
<th>Function</th>
<th>Report the available buffer size.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>SIZE</td>
</tr>
</tbody>
</table>

**TYPE**

<table>
<thead>
<tr>
<th>Function</th>
<th>Return UTC module number, and version description.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>TYPE</td>
</tr>
</tbody>
</table>

**VER**

<table>
<thead>
<tr>
<th>Function</th>
<th>Report the controller firmware version.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>VER</td>
</tr>
</tbody>
</table>
Chapter 2. Designing With UTC Control

In this chapter, we will study how to start a project design with UTC control. We will start with some simple application projects that are ready for applying to a customer’s machine. All the sample projects include a series of illustrations of the design procedures. Before going through with this chapter, please at least have a rough review on “UTC Motion Controller Manual” to get the basic idea of all the variables, parameters, online commands and buffer commands.

2.1 A cut to length with timer cut tool output control

This is a single axis servo control system project. When the system auto-cycle stops, some manual control is allowed. When the system auto-cycle starts, the control will go over the sequences described as following. The auto cycle can be terminated by a stop button.

1. The cut tool output turns on for the 1st pre-set timing.
2. When the 1st setting timer is over, the cut tool turns off and starts the 2nd pre-set timer.
3. When the 2nd timer timeout, servomotor feed a pre-set length.
4. When the servomotor finishes the feed length, the cutting counter increase by one and the control go back to sequence 1.
5. If the cutting count over a pre-set limit, the control stops feeding.

2.1.1 The details of project requirement

2.1.1.1 Parameter Settings

1. Length setting --- ---
   Basic unit: 0.1 mm
   Setting range: 0 to 99999.9 mm
2. Speed setting ---
   Setting range: 1 mm/second to 999mm/second
3. Count limit setting ---
   Setting range: 0 to 999999; zero setting indicates unlimited cutting.
4. Cut tool on time setting (timer1): 0.1 seconds to 99.9 seconds
5. Cut tool off time setting (timer2): 0.1 seconds to 9.9 seconds
6. Machine gear ratio: encoder counts per basic unit (0.1 mm)
7. Acceleration/deceleration time setting. 100msec to 999 msec
2.1.1.2 Control System Consideration

1. Parameter settings will be done with a customer-designed panel to run with UtcPanelEng application software. Please read the “PMWIN Manual” to get the idea of the user panel design.

2. The manual control keys.
   The manual control buttons are placed on the computer screen.
   As well, there are some digital inputs are ready for external control connections.

3. Cut tool output.
   One digital output to drive a solid-state relay as an interface to the control valve.

4. Servomotor control
   Use UTC controller Driver1 port to connect with a servo driver.

2.1.2 Start the Design

2.1.2.1 Hardware Assignments

1. Digital inputs assignments
   M11: START --- This input signal (M11=0) to trigger the system to start auto cycle.
   M12: STOP --- This input signal (M12=0) to stop the auto cycle.
   M13: JOG + --- This input signal to jog the motor in positive direction at stop cycle.
   M14: JOG- --- This input signal to jog the motor in negative direction at stop cycle.
   M15: Clear --- This input signal to clear the current cutting counter to zero.

2. Digital output assignments
   M01: Cut Tool --- M01=0 (LED lit) to turn on the cut tool.

2.1.2.2 Internal Variable Assignments

1. Setting parameters variable assignments
   Cutting length: P400
   Speed setting: P401
   Count limit: P402
   Tool on time (timer1): P403
   Tool off time (timer2): P404

2. Display and control information variable assignments
   Accumulated cut counts: P410
   Current servo position: P411
   Current status: P412
3. System configure and protection variables
   Gear ratio: P420 (Encoder counts per basic unit, 0.1mm)
   Maximum speed: P421 counts/second
   Minimum acceleration time: P422 msec
   Error message: P423

2.1.2.3 PLC programs planning example
1. PLC0 foreground (not used)
2. PLC1: Power up initial settings
3. PLC2: PLC enable/disable management
4. PLC3: Individual key/push-button handling
5. PLC4: Calculations
6. PLC5: Auto cycle handling
7. PLC6: Alarm/protection handling
8. PLC7: External inputs handling

2.1.2.4 PLC and motion programs coding example
1. PLC program
   Open plc1 clear ;open a buffer area for plc1 and clear that area
   ...... ;the program content
   close ;close plc1 area

2. Motion program
   Open prog1 clear ;open a buffer area for motion program1
   ...... ;and clear that area
   ...... ;the program content
   close ;close program 1 area
2.2 A cut to length with two sensors cut tool output control, mark signal capture and Hitech ADP3 as HMI

This project is similar to the first project in section 2.1. We add a signal capture option and also change the HMI design to fit Hitech ADP3. We can learn how to fit the Hitech HMI to UTC control.

The following portion is different from the project in section 2.1.

- When the cut tool output turns on, a lower limit sensor to trigger the cut tool off.
- When the cut tool output turns off, an upper limit sensor to trigger next feeding.
- Before the setting length feeding, the servomotor speed up and catch a mark signal to determine the feeding length. The total length will be as length before mark plus setting length.
- The Hitech ADP3 will be the target HMI.

1. The cut tool output turns on until a lower limit switch on. If the 1st pre-set timer timeout, the control generate an error message, cutter on timeout.
2. When the lower sensor turns on, the cut tool turns off and looks for the upper sensor turns on. If the sensor does not turn on before timer2 timeout, the control generate an error message, cutter off timeout.
3. When the upper sensor turns on, the servomotor keeps feeding and looks for a mark signal to continue to feed the pre-set length.
4. When the servomotor finishes the pre-set feed length, the cutting counter increase by one and the control go back to sequence 1.
5. If the cutting count over a pre-set limit, the control stops feeding.

2.2.1 The details of project requirement

2.2.1.1 Parameter Settings

1. Length setting --- ---
   - Basic unit: 0.1 mm
   - Setting range: 0 to 99999.9 mm
2. Speed setting ---
   - Setting range: 1 mm/second to 999 mm/second
3. Count limit setting ---
   - Setting range: 0 to 999999; zero setting indicates unlimited cutting.
4. Cut tool on limit timer (timer1): 0.1 seconds to 99.9 seconds
5. Cut tool off limit timer (timer2): 0.1 seconds to 9.9 seconds
6. Machine gear ratio: encoder counts per basic unit (0.1 mm)
7. Acceleration/deceleration time setting. 100msec to 999 msec
2.2.1.2 Control System Consideration

1. Parameter settings will be done with Hitech ADP3 HMI.
2. The panel control keys and push-buttons are on Hitech touch panel. As well, there are some digital inputs ready for external control connections.
3. Cut tool output.
   One digital output is assigned to drive a solid-state relay as an interface to the control valve.
4. Servomotor control
   Use UTC controller Driver1 port to connect with a servo driver.

2.2.2 Start the Design

2.2.2.1 Hardware Assignments

1. Digital inputs assignments
   M11: START --- This input signal (M11=0) to trigger the system to start auto cycle.
   M12: STOP --- This input signal (M12=0) to stop the auto cycle.
   M13: JOG + --- This input signal to jog the motor in positive direction at stop cycle.
   M14: JOG- --- This input signal to jog the motor in negative direction at stop cycle.
   M15: Clear --- This input signal to clear the current cutting counter to zero.
   M21: Lower limit sensor
   M22: Upper limit sensor
   M23: Mark signal to trigger the pre-set length.

2. Digital output assignments
   M01: Cut Tool --- M01=0 (LED lit) to turn on the cut tool.

2.2.2.2 Internal Variable Assignments

1. Setting parameters variable assignments
   Cutting length: P400
   Speed setting: P401
   Count limit: P402
   Tool on timer (timer1): P403
   Tool off timer (timer2): P404

2. Display and control information variable assignments
   Accumulated cut counts: P410
   Current servo position: P411
   Current status: P412
3. System configure and protection variables
   Gear ratio: P420 (Encoder counts per basic unit, 0.1mm)
   Maximum speed: P421 counts/second
   Minimum acceleration time: P422 msec
   Error message: P423

4. Hitech interface parameters
   Cutting length: P500, P501
   Speed setting: P502
   Count limit: P503, P504
   Tool on timer (timer1): P505
   Tool off timer (timer2): P506
   Accumulated cut counts: P507, P508
   Current servo position: P509, P510
   Current status: P511

   Gear ratio: P512
   Maximum speed: P513, P514
   Error message: P423

2.2.2.3 PLC programs planning example

1. PLC0: Mark capture handling
2. PLC1: Power up initial settings
3. PLC2: PLC enable/disable management
4. PLC3: Individual key/push-button handling
5. PLC4: Calculations
6. PLC5: Auto cycle handling
7. PLC6: Alarm/protection handling
8. PLC7: External inputs handling
2.3 A single axis motion program example for drilling cycle and Hitech ADP3 as HMI

(Home finding // limit switches // motion program for the drilling // hand wheel for master following setting // AD1 for federate override // AD2 and DA1 for spindle speed control)
Chapter 3. Appendix

3.1 PROJECT_1

CLOSE

;==========EXTERNAL INPUT==========
;STATUS =0 (TRUE)
; =1 (FALSE)
;M11 ;START
;M12 ;STOP
;M13 ;+JOG
;M14 ;-JOG
;M15 ;CLEAR

;==========EXTERNAL INPUT==========
;STATUS =0 (TRUE)
; =1 (FALSE)
;M21 ;NO MATERIAL

;==========OUTPUT==========
;STATUS =0 (TRUE)
; =1 (FALSE)
;M01 ;CUT TOOL
;M02 ;ALARM LAMP

;==========SETTING PARAMETERS====
;P400 ;CUTTING LENGTH SETTING
;P401 ;SPEED SETTING
;P402 ;COUNT LIMIT SETTING
;P403 ;TOOL ON TIME SETTING
;P404 ;TOOL OFF TIME SETTING
;P405 ;ACC/DEC TIME SETTING

;==========DISPLAY===========
;P410 ;ACCUMULATED CUT COUNTS
;P411 ;CURRENT SERVO POSITION

;==========SYSTEM CONFIGURE===
;P420 ;GEAR RATIO
;P421 ;MAXIMUM SPEED
;P422 ;MINIMUM ACCELERATION TIME
;P423 ;ERROR MESSAGE
;P424 ;ACTION CODE
;P425 ;AUTO RUN STATUS
;P426 ;FEEDING FLAG
;P427 ;COUNT OVER FLAG

;==================PLC0==================
OPEN PLC0 CLEAR
CLOSE

;==================PLC1==================
OPEN PLC1 CLEAR ;POWER UP INITIAL SETTINGS
DISPLC2,3,4,5,6,7
M71=2000
WHILE(M71>0) ;DELAY 2 SECOND
ENDW
P11=M11 ;START FLAG
P12=M12 ;STOP FLAG
P13=M13 ;+JOG FLAG
P14=M14 ;-JOG FLAG
P15=M15 ;CLEAR FLAG
P424=0 ;NO ACTIONS AT POWER ON
P425=0 ;NOT IN AUTO RUN STATUS
P426=0 ;NOT IN AUTO FEEDING
P427=0 ;NO COUNT OVER
P2=0 ;P2=1 WHEN AUTO RUNNING
CMD"#1J/" ;ENABLE #1 SEVRO ON
ENAPLC2,4,6,7
DISPLC1
CLOSE

;==================PLC2==================
OPEN PLC2 CLEAR ;PLC ENABLE/DISABLE MANAGEMENT
IF(P424!=0) ;ANY ACTIONS?
    ENAPLC3 ;IF YES, ENABLE PLC3.
ENDIF
IF(P425!=0) ;AUTO CYCLE STARTED?
    ENAPLC5 ;IF YES, ENABLE PLC5.
ENDIF
CLOSE
OPEN PLC3 CLEAR ; INDIVIDUAL KEY/PUSH-BUTTON HANDING

IF(P424=1) ; JOG+
P424=0
IF(P425=0)
  I101=P405 ; ACC/DEC TIME SETTING
  I103=P401*M191 ; JOG SPEED SETTING
  CMD"#1J+
  P426=0 ; NOT IN AUTO FEEDING
ENDIF

ENDIF

IF(P424=2) ; JOG-
P424=0
IF(P425=0)
  I101=P405
  I103=P401*M191
  CMD"#1J-
  P426=0
ENDIF

ENDIF

IF(P424=3) ; JOG/
P424=0
IF(P425=0 AND M144=0) ; NOT AUTO & NO FOLLOWING ERROR
  CMD"#1J/
  P426=0
ENDIF

ENDIF

IF(P424=4) ; CLEAR
P424=0
P410=0 ; CLEAR CUT COUNTS
ENDIF

IF(P424=10) ; START
  P2=1 ; STARTED FLAG
P424=0
  IF(P425=0 AND M141=0 AND P423=0)
    IF(P426=0)
P425=1 ; CUT ONCE THEN MOVE

ELSE

P425=3 ; BACK TO UNFINISHED MOVING

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

IF(P424=11) ; STOP

P2=0 ; STOP AUTO CYCLE

P424=0

IF(P425<>0) ; IF IN AUTO CYCLE.

P425=10 ; TO STOP PLC5

ELSE

CMD"#1J/" ; STOP JOGGING

ENDIF

ENDIF

DISPLC3 ; DISABLE PLC3

CLOSE

; = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

OPEN PLC4 CLEAR ; CALCULATIONS

P411=ROUND(M161/M191) ; CURRENT SERVO POSITION

IF(P400<0 OR P400>99999.9) ; LENGTH SETTING RANGE

IF(P400>99999.9)

P400=99999.9

ELSE

P400=0

ENDIF

ENDIF

IF(P400<>Q400) ; CHECK LENGTH SETTING

P426=0

Q400=P400

ENDIF

IF(P401<1 OR P401>P421) ; SPEED SETTING RANGE

IF(P401>P421) ; MAXIMUM SPEED FOR P421

P401=P421

ELSE

P401=1
ENDIF
ENDIF
IF(P402<0 OR P402>999999) ;COUNT LIMIT SETTING RANGE
  IF(P402>999999)
    P402=999999
  ELSE
    P402=0
  ENDIF
ENDIF
IF(P403<0.1 OR P403>99.9) ;TOOL ON TIMER RANGE
  IF(P403>99.9)
    P403=99.9
  ELSE
    P403=0.1
  ENDIF
ENDIF
IF(P404<0.1 OR P404>99.9) ;TOOL OFF TIMER RANGE
  IF(P404>99.9)
    P404=99.9
  ELSE
    P404=0.1
  ENDIF
ENDIF
IF(P405<P422 OR P405>999) ;ACCELERATION/DECELERTION ;TIMER RANGE
  IF(P405>999) ;MINIMUM ACC/DEC TIMER FOR P422
    P405=999
  ELSE
    P405=P422
  ENDIF
ENDIF
IF(P420<=0) ;GEAR RATIO
  P420=1
ENDIF
M191=P420*10
CLOSE
OPEN PLC5 CLEAR ; AUTO CYCLE HANDLING
IF(P425=1) ; FIRST TOOL ON
   M1=0 ; TOOL ON
   M71=P403*1000 ; SET TOOL ON TIME
ENDW
WHILE(M71>0)
   ENDW
P425=2
ENDIF

IF(P425=2) ; FIRST TOOL OFF
   M1=1
   M71=P404*1000 ; SET TOOL OFF TIME BEFORE FEEDING
ENDW
WHILE(M71>0 AND P425=2)
   ENDW
IF(P425=10)
   RET
ENDIF
P425=3
ENDIF

IF(P425=3) ; FEEDING
IF(P426=0)
   CMD"#1HMZ" ; SET CURRENT POSITION = 0
ENDIF
M71=10
WHILE(M71>0) ; WAIT FOR CLEAR POSITION
   ENDW
M163=M191*P400 ; M163 = TARGET POSITION
I101=P405 ; ACC/DEC TIME
I103=P401*M191 ; SPEED
CMD"#1J=" ; ACTION
M71=20
WHILE(M71>0) ; WAIT FOR FIRMWARE PROCESSING
   ENDW
P426=1 ; FEEDING FLAG = 1
WHILE(M161!=M163 AND M141!=0 AND P425=3)
   ENDW
IF(P425=10) ; STOP
RET
ENDIF
P426=0 ;STOP FEEDING
P425=4 ;NEXT STEP
ENDIF
IF(P425=4) ;TOOL ON
  M1=0
  M71=P403*1000
  WHILE(M71>0)
  ENDW
  P425=5
ENDIF
IF(P425=5) ;TOOL OFF
  M1=1
  M71=P404*1000
  WHILE(M71>0 AND P425=5)
  ENDW
  IF(P425=10)
    RET
  ENDIF
  P410=P410+1 ;COUNT
  IF(P402>P410 OR P402=0) ;CHECK COUNT LIMIT
    P425=3
  ELSE
    P423=4 ;ALARM MESSAGE FOR P423=3
    P425=10
  ENDIF
ENDIF
IF(P425=10) ;GET READY FOR STOP
  CMD"#1J/"
  M1=1
  P425=0
ENDIF
IF(P425=0) ;DISABLE PLC5 TO STOP AUTO CYCLE
  DISPLC5
ENDIF
CLOSE

........................................
OPEN PLC6 CLEAR                    ;ALARM/PROTECTION HANDLING
IF(P402>P410 OR P402=0)            ;COUNT OVER FLAG
    P427=0
ELSE
    P427=1
ENDIF
IF(M143=1 OR M21=0 OR M144=1)     ;DRIVER FAULT OR NO MATERIAL OR
    ;FOLLOWING ERROR
    IF(M143=1)
        P423=1
    ELSE
        IF(M21=0)
            P423=2
        ELSE
            P423=3
        ENDIF
    ENDIF
ELSE
    IF(P427=1)                      ;COUNT OVER
        P423=4
    ELSE
        P423=0
    ENDIF
ENDIF
                    ..................................................
        IF(P423!=0)
            M2=P2^1
            P430=P423
        ELSE
            M2=1
            IF(P425=0 AND P426=1)
                P430=29
            ELSE
                P430=20+P425
            ENDIF
        ENDIF
                    ..................................................
        CLOSE
;==============PLC7==============
OPEN PLC7 CLEAR ;EXTERNAL HMI INPUTS
IF(M11=0 AND P11!=0) ;START EXTERNAL INPUTS
    P11=0
    P424=10
ENDIF
IF(M11=1 AND P11!=1)
    P11=1
ENDIF
IF(M12=0 AND P12!=0) ;STOP EXTERNAL INPUTS
    P12=0
    P424=11
ENDIF
IF(M12=1 AND P12!=1)
    P12=1
ENDIF
IF(M13=0 AND P13!=0) ;JOG+ EXTERNAL INPUTS
    P13=0
    P424=1
ENDIF
IF(M13=1 AND P13!=1) ;JOG/ EXTERNAL INPUTS
    P13=1
    P424=3
ENDIF
IF(M14=0 AND P14!=0) ;JOG- EXTERNAL INPUTS
    P14=0
    P424=2
ENDIF
IF(M14=1 AND P14!=1) ;JOG/ EXTERNAL INPUTS
    P14=1
    P424=3
ENDIF
IF(M15=0 AND P15!=0) ;CLEAR EXTERNAL INPUTS
    P15=0
    P424=4
ENDIF
IF(M15=1 AND P15!=1)
    P15=1
ENDIF
CLOSE

;================PLC8================
OPEN PLC8 CLEAR
CLOSE
3.2 PROJECT_2

CLOSE

;=================EXTERNAL INPUT=================
;STATUS =0 (TRUE)
 ; =1 (FALSE)
;M11 ;START
;M12 ;STOP
;M13 ;+JOG
;M14 ;-JOG
;M15 ;CLEAR

;=================EXTERNAL INPUT=================
;STATUS =0 (TRUE)
 ; =1 (FALSE)
;M21 ;NO MATERIAL
;;M22 ;CUT UP LIMIT
;;M23 ;CUT DOWN LIMIT

;=================PANEL INPUT=================
;STATUS =0 (FALSE)
 ; =1 (TRUE)
M501..505->*
;;M501 ;+JOG
;;M502 ;-JOG

;=================OUTPUT==============
;STATUS =0 (TRUE)
 ; =1 (FALSE)
;M01 ;CUT TOOL
;M02 ;ALARM LAMP

;=================SETTING PARAMETERS=====
;P400 ;CUTTING LENGTH<*****.*>
;P401 ;SPEED SETTING<***>
;P402 ;COUNT LIMIT<******>
;P403 ;TOOL ON TIME(TIMER1)<**.*>
;P404 ;TOOL OFF TIME<**.*>
;P405  ;ACC/DEC TIMER<***>
;P406  ;TOOL TO LIMIT TIME OUT<***>

;==========DISPLAY==========
;P410  ;ACCUMULATED CUT COUNTS<******>
;P411  ;CURRENT SERVO POSITION<*****.*>
;P412  ;CURRENT STATUS

;==========SYSTEM CONFIGURE===
;P420  ;GEAR RATIO<****.***>
;P421  ;MAXIMUM SPEED<***>
;P422  ;MINIMUM ACCELERATION TIME<***>
;P423  ;ERROR MESSAGE<1~60>
;P424  ;ACTION STATUS<1~20>
;P425  ;AUTO RUN STATUS<1~5>
;P426  ;FEEDING FLAG<0,1>
;P427  ;COUNT OVER FLAG<0,1>
;P428  ;TOOL ON TIME OUT FLAG<0,1>
;P429  ;TOOL OFF TIME OUT FLAG<0,1>

;P430  ;INFORMATION

;==========HUMAN MACHINE INTERFACE=========
;;;;;;READ UTC-CONTROLLER;;;;;;
;P450  ;P400---
;P451  ;P401(SPEED SETTING)
;P452  ;P402---
;P453  ;P403(TOOL ON TIME)
;P454  ;P404(TOOL OFF TIME)
;P455  ;P405(ACC/DEC TIMER)
;P456  ;P406(TOOL TO LIMIT TIME OUT)

;P460  ;P410---
;P461  ;P411---

;P470  ;P420---
;P471  ;P421(MAXIMUM SPEED)
;P472  ;P422(MINIMUM ACCELERATION TIME)
;P480 ;LOW_Word FOR P400(CUTTING LENGTH)
;P481 ;HI_Word FOR P400(CUTTING LENGTH)
;P482 ;LOW_Word FOR P402(COUNT LIMIT)
;P483 ;HI_Word FOR P402(COUNT LIMIT)
;P484 ;LOW_Word FOR P410(ACCUMULATED CUT COUNTS)
;P485 ;HI_Word FOR P410(ACCUMULATED CUT COUNTS)
;P486 ;LOW_Word FOR P411(CURRENT SERVO POSITION)
;P487 ;HI_Word FOR P411(CURRENT SERVO POSITION)
;P488 ;LOW_Word FOR P420(GEAR RATIO)
;P489 ;HI_Word FOR P420(GEAR RATIO)

;;;;;;WRITE TO UTC-CONTROLLER;;;;;;

;P550 ;P400---
;P551 ;P401(SPEED SETTING)
;P552 ;P402---
;P553 ;P403(TOOL ON TIME)
;P554 ;P404(TOOL OFF TIME)
;P555 ;P405(ACC/DEC TIMER)
;P556 ;P406(TOOL TO LIMIT TIME OUT)

;P560 ;P410---
;P561 ;P411---

;P570 ;P420---
;P571 ;P421(MAXIMUM SPEED)
;P572 ;P422(MINIMUM ACCELERATION TIME)

;P580 ;LOW_Word FOR P400(CUTTING LENGTH)
;P581 ;HI_Word FOR P400(CUTTING LENGTH)
;P582 ;LOW_Word FOR P402(COUNT LIMIT)
;P583 ;HI_Word FOR P402(COUNT LIMIT)
;P584 ;LOW_Word FOR P410(ACCUMULATED CUT COUNTS)
;P585 ;HI_Word FOR P410(ACCUMULATED CUT COUNTS)
;P586 ;LOW_Word FOR P411(CURRENT SERVO POSITION)
;P587 ;HI_Word FOR P411(CURRENT SERVO POSITION)
;P588 ;LOW_Word FOR P420(GEAR RATIO)
;P589 ;HI_Word FOR P420(GEAR RATIO)
;=============PLC0==================
OPEN PLC0 CLEAR
CLOSE

;=============PLC1==================
OPEN PLC1 CLEAR ;POWER UP INITIAL SETTINGS
DISPLC2,3,4,5,6,7,8
M71=2000
WHILE(M71>0) ;DELAY 2 SECOND
ENDW
CMD"#1J/";ENABLE #1 SEVRO ON
P11=M11 ;START FLAG
P12=M12 ;STOP FLAG
P13=M13 ;+JOG FLAG
P14=M14 ;-JOG FLAG
P15=M15 ;CLEAR FLAG
P501=M501 ;START FLAG
P502=M502 ;STOP FLAG
P503=M503 ;+JOG FLAG
P504=M504 ;-JOG FLAG
P505=M505 ;CLEAR FLAG
P424=0 ;NO ACTIONS AT POWER ON
P425=0 ;NOT IN AUTO RUN STATUS
P426=0 ;NOT IN AUTO FEEDING
P427=0
P428=0
P429=0
P2=0 ;P2=1 WHEN AUTO RUNNING
CMD"P550..599=0"
ENAPLC2,4,6,7,8
DISPLC1
CLOSE

;=============PLC2==================
OPEN PLC2 CLEAR ;PLC ENABLE/DISABLE MANAGEMENT
IF(P424!=0) ;ANY ACTIONS?
    ENAPLC3 ;IF YES, ENABLE PLC3.
ENDIF
IF(P425!=0) ;AUTO CYCLE STARTED?
    ENAPLC5 ;IF YES, ENABLE PLC5.
ENDIF
CLOSE

;=============PLC3=============
OPEN PLC3 CLEAR ;INDIVIDUAL KEY/PUSH-BUTTON HANDING
IF(P424=1) ;JOG+
P424=0
    IF(P425=0)
        I101=P405 ;ACC/DEC TIME SETTING
        I103=P401*M191 ;JOG SPEED SETTING
        CMD"#1J+
        P426=0 ;NOT IN AUTO FEEDING
    ENDF
ENDIF
ENDF
IF(P424=2) ;JOG-
P424=0
    IF(P425=0)
        I101=P405
        I103=P401*M191
        CMD"#1J-
        P426=0
    ENDF
ENDIF
IF(P424=3) ;JOG/
P424=0
    IF(P425=0)
        CMD"#1J/
        ;STOP JOGGING
    ENDF
ENDIF
IF(P424=4) ;CLEAR
    P424=0
    P410=0 ;CLEAR CUT COUNTS
    P428=0 ;;
    P429=0 ;;
ENDIF
IF(P424=10) ;START
    P2=1 ;STARTED FLAG
P424=0
IF(P425=0 AND M141=0 AND P423=0)
    IF(P426=0)
        P425=1 ;CUT ONCE THEN MOVE
    ELSE
        P425=3 ;BACK TO UNFINISHED MOVING
    ENDIF
ENDIF
ENDIF
IF(P424=11) ;STOP
    P2=0 ;STOP AUTO CYCLE
    P424=0
ENDIF
IF(P425!=0) ;IF IN AUTO CYCLE.
    P425=10 ;TO STOP PLC5
ELSE
    P423=0
    P427=0
    P428=0
    P429=0
    CMD"#1J/" ;STOP JOGGING
ENDIF
DISPLC3 ;DISABLE PLC3
CLOSE

;=============PLC4=============
OPEN PLC4 CLEAR ;CALCULATIONS
P411=ROUND(M161/M191) ;CURRENT SERVO POSITION
IF(P400<0 OR P400>99999.9) ;LENGTH SETTING RANGE
    IF(P400>99999.9)
        P400=99999.9
    ELSE
        P400=0
    ENDIF
ENDIF
IF(P400!=Q400) ;CHECK LENGTH SETTING
    P426=0
    Q400=P400
ENDIF
IF(P401<1 OR P401>P421) ;SPEED SETTING RANGE
    IF(P401>P421) ;MAXIMUM SPEED FOR P421
        P401=P421
    ELSE
        P401=1
    ENDIF
ENDIF
IF(P402<0 OR P402>999999) ;COUNT LIMIT SETTING RANGE
    IF(P402>999999)
        P402=999999
    ELSE
        P402=0
    ENDIF
ENDIF
IF(P403<0.1 OR P403>99.9) ;TOOL ON TIMER RANGE
    IF(P403>99.9)
        P403=99.9
    ELSE
        P403=0.1
    ENDIF
ENDIF
IF(P404<0.1 OR P404>99.9) ;TOOL OFF TIMER RANGE
    IF(P404>99.9)
        P404=99.9
    ELSE
        P404=0.1
    ENDIF
ENDIF
IF(P405<P422 OR P405>999) ;ACCELERATION/DECELERATION
    IF(P405>999) ;MINIMUM ACC/DEC TIMER FOR P422
        P405=999
    ELSE
        P405=P422
    ENDIF
ENDIF
IF(P420<1) ;GEAR RATIO
    P420=1
M191=P420*10
CLOSE

;=============PLC5=============
OPEN PLC5 CLEAR ;AUTO CYCLE HANDLING
IF(P425=1) ;FIRST TOOL ON
M1=0 ;TOOL ON
M71=P406*1000 ;TO DOWN LIMIT TIME
WHILE(M71>0 AND P425=1 AND M23=1)
ENDW
IF(M71<0) ;TIME OUT
P428=1
P425=10
RET
ENDIF
IF(P425=10) ;STOP
RET
ENDIF
M71=P403*1000 ;SET TOOL ON TIME
WHILE(M71>0)
ENDW
IF(P425=10) ;STOP
RET
ENDIF
P425=2
ENDIF
IF(P425=2) ;FIRST TOOL OFF
M1=1
M71=P406*1000 ;TOOL OFF TO UP LIMIT TIME
WHILE(M71>0 AND P425=2 AND M22=1)
ENDW
IF(M71<0) ;TIME OUT
P429=1
P425=10
RET
ENDIF
IF(P425=10)
RET
ENDIF
M71=P404*1000 ;SET TOOL OFF TIME BEFORE FEEDING
WHILE(M71>0 AND P425=2)
ENDW
IF(P425=10)
    RET
ENDIF
P425=3
ENDIF
IF(P425=3) ;FEEDING
    IF(P426=0)
        CMD"#1HMZ" ;SET CURRENT POSITION = 0
    ENDIF
    M71=10
    WHILE(M71>0) ;WAIT FOR CLEAR POSITION
        ENDW
    M163=M191*P400 ;M163 = TARGET POSITION
    I101=P405 ;ACC/DEC TIME
    I103=P401*M191 ;SPEED
    CMD"#1J=" ;ACTION
    M71=20
    WHILE(M71>0) ;WAIT FOR FIRMWARE PROCESSING
        ENDW
    P426=1 ;FEEDING FLAG = 1
WHILE(M161!=M163 AND M141!=0 AND P425=3)
ENDW
IF(P425=10) ;STOP
    RET
ENDIF
P426=0 ;STOP FEEDING
P425=4 ;NEXT STEP
ENDIF
IF(P425=4) ;TOOL ON
    M1=0
    M71=P406*1000
    WHILE(M71>0 AND P425=4 AND M23=1)
        ENDW
    IF(M71<0) ;TIME OUT
        P428=1
56
P425=10
RET
ENDIF
IF(P425=10) ;STOP
   RET
ENDIF
M71=P403*1000 ;SET TOOL ON TIME
WHILE(M71>0)
   ENDW
IF(P425=10) ;STOP
   RET
ENDIF
P425=5
ENDIF
IF(P425=5) ;TOOL OFF
   M1=1
   M71=P406*1000
   WHILE(M71>0 AND P425=5 AND M22=1)
      ENDW
   IF(M71<0) ;TIME OUT
      P429=1
      P425=10
      RET
   ENDIF
   IF(P425=10) ;STOP
      RET
   ENDIF
M71=P404*1000 ;SET TOOL OFF TIME BEFORE FEEDING
WHILE(M71>0 AND P425=5)
   ENDW
IF(P425=10)
   RET
ENDIF
P410=P410+1 ;COUNT
IF(P402>P410 OR P402=0) ;CHECK COUNT LIMIT
   P425=3
ELSE
   P423=3 ;ALARM MESSAGE FOR P423=3
   P425=10
ENDIF
ENDIF
ENDIF
IF(P425=10)  ;GET READY FOR STOP
  IF(M114=0)
    CMD"#1J/
  ENDIF
  M1=1
  P425=0
ENDIF
IF(P425=0)  ;DISABLE PLC5 TO STOP AUTO CYCLE
  DISPLC5
ENDIF
CLOSE

;=============PLC6=============
OPEN PLC6 CLEAR  ;ALARM/PROTECTION HANDING
IF(P402>P410 OR P402=0)  ;COUNT OVER FLAG
  P427=0
ELSE
  P427=1
ENDIF
IF(M143=1 OR M21=0 OR M144=1)  ;DRIVER FAULT OR NO MATERIAL OR 
  ;FOLLOWING ERROR
  IF(M143=1)
    P423=1
  ELSE
    IF(M21=0)
      P423=2
    ELSE
      P423=3
    ENDIF
  ENDIF
ELSE
  IF(P427=1 OR P428=1 OR P429=1)
    IF(P427=1)
      P423=4
    ELSE
      IF(P428=1)
        P423=5
      ELSE
        P423=5
      ENDIF
    ENDIF
  ELSE
    P423=5
  ENDIF
ELSE
  P423=6
ENDIF
ENDIF
ENDIF
ENDIF
IF(P423!=0)
  M2=P2^1
  P430=P423
  IF(P2=1)
    P424=11 ;STOP
    P426=0
  ENDIF
ELSE
  M2=1
  IF(P425=0 AND P426=1)
    P430=29
  ELSE
    P430=20+P425
  ENDIF
ENDIF
CLOSE
;=============PLC7=============OPEN PLC7 CLEAR ;EXTERNAL&PANEL INPUTS
IF(M11=0 AND P11!=0) ;START EXTERNAL INPUTS
  P11=0
  P424=10
ENDIF
IF(M11=1 AND P11!=1)
  P11=1
ENDIF
IF(M12=0 AND P12!=0) ;STOP EXTERNAL INPUTS
  P12=0
  P424=11
ENDIF
IF(M12=1 AND P12!=1)
  P12=1
ENDIF
IF(M13=0 AND P13!=0) ; JOG+ EXTERNAL INPUTS
  P13=0
  P424=1
ENDIF ; JOG/ EXTERNAL INPUTS
IF(M13=1 AND P13!=1)
  P13=1
  P424=3
ENDIF
IF(M14=0 AND P14!=0) ; JOG- EXTERNAL INPUTS
  P14=0
  P424=2
ENDIF
IF(M14=1 AND P14!=1) ; JOG/ EXTERNAL INPUTS
  P14=1
  P424=3
ENDIF
IF(M15=0 AND P15!=0) ; CLEAR EXTERNAL INPUTS
  P15=0
  P424=4
ENDIF
IF(M15=1 AND P15!=1)
  P15=1
ENDIF
..............................
IF(M501=1 AND P501!=1) ; JOG+ PANEL INPUTS
  P501=1
  P424=1
ENDIF ; JOG/ PANEL INPUTS
IF(M501=0 AND P501!=0)
  P501=0
  P424=3
ENDIF
IF(M502=1 AND P502!=1) ; JOG- PANEL INPUTS
  P502=1
  P424=2
ENDIF
IF(M502=0 AND P502!=0) ; JOG/ PANEL INPUTS
  P502=0
  P424=3
ENDIF
CLOSE

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
OPEN PLC8 CLEAR
;;;;;;READ UTC-CONTROLLER;;;;;;

;P450  ;P400---
P451=P401                ;P401(SPEED SETTING)
P452=P402--
P453=P403*10            ;P403(TOOL ON TIME)
P454=P404*10            ;P404(TOOL OFF TIME)
P455=P405                ;P405(ACC/DEC TIMER)
P456=P406                ;P406(TOOL TO LIMIT TIME OUT)

;P460  ;P410---
P461  ;P411---

;P470  ;P420---
P471=P421                ;P421(MAXIMUM SPEED)
P472=P422                ;P422(MINIMUM ACCELERATION TIME)

P481=INT(P400*10/65536)        ;HI_Word FOR P400(CUTTING LENGTH)
P480=INT(P400*10)-P481*65536   ;LOW_Word FOR P400(CUTTING LENGTH)
P483=INT(P402/65536)          ;HI_Word FOR P402(COUNT LIMIT)
P482=P402-P483*65536          ;LOW_Word FOR P402(COUNT LIMIT)
P485=INT(P410/65536)          ;HI_Word FOR P410(ACCUMULATED CUT COUNTS)
P484=P410-P485*65536          ;LOW_Word FOR P410(ACCUMULATED CUT COUNTS)
P487=INT(P411*10/65536)       ;HI_Word FOR P411(CURRENT SERVO POSITION)
P486=INT(P411*10)-P487*65536   ;LOW_Word FOR P411(CURRENT SERVO POSITION)
P489=INT(P420*1000/65536)     ;HI_Word FOR P420(GEAR RATIO)
P488=INT((P420+0.0001)*1000)-P489*65536  ;LOW_Word FOR P420(GEAR RATIO)

;;;;;;WRITE TO UTC-CONTROLLER;;;;;;

;P550  ;P400---
IF(P551!=0)                ;P401(SPEED SETTING)
P401=P551
P551=0
ENDIF

;P552 ;P402---
IF(P553!=0) ;P403(TOOL ON TIME)
    P403=P553*0.1
    P553=0
ENDIF

IF(P554!=0) ;P404(TOOL OFF TIME)
    P404=P554*0.1
    P554=0
ENDIF

IF(P555!=0) ;P405(ACC/DEC TIMER)
    P405=P555
    P555=0
ENDIF

IF(P556!=0) ;P406(TOOL TO LIMIT TIME OUT)
    P406=P556
    P556=0
ENDIF

;P560 ;P410---
;P561 ;P411---

;P570 ;P420---
IF(P571!=0) ;P421(MAXIMUM SPEED)
    P421=P571
    P571=0
ENDIF

IF(P572!=0) ;P422(MINIMUM ACCELERATION TIME)
    P422=P572
    P572=0
ENDIF

;P580 ;LOW_WORD FOR P400(CUTTING LENGTH)
;P581 ;HI_WORD FOR P400(CUTTING LENGTH)
IF(P580!=0 OR P581!=0)
    P400=(P580+P581*65536)*0.1
    P580=0
ENDIF
P581=0

ENDIF

;P582 ;LOW_WORD FOR P402(COUNT LIMIT)
;P583 ;HI_WORD FOR P402(COUNT LIMIT)
IF(P582!=0 OR P583!=0)
    P402=P582+P583*65536
    P582=0
    P583=0
ENDIF

;P584 ;LOW_WORD FOR P410(ACCUMULATED CUT COUNTS)
;P585 ;HI_WORD FOR P410(ACCUMULATED CUT COUNTS)
IF(P584!=0 OR P585!=0)
    P410=P584+P585*65536
    P584=0
    P585=0
ENDIF

;P586 ;LOW_Word FOR P411(CURRENT SERVO POSITION)
;P587 ;HI_Word FOR P411(CURRENT SERVO POSITION)
IF(P586!=0 OR P587!=0)
    P411=(P586+P587*65536)*0.1
    P586=0
    P587=0
ENDIF

;P588 ;LOW_word FOR P420(GEAR RATIO)
;P589 ;HI_Word FOR P420(GEAR RATIO)
IF(P588!=0 OR P589!=0)
    P420=(P588+P589*65536)*0.001
    P588=0
    P589=0
ENDIF

CLOSE