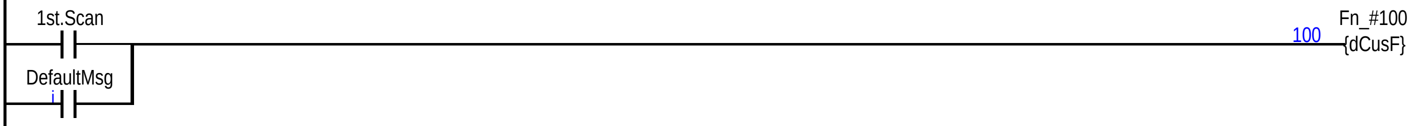
1

Driving an Optimate KM420 HMI Panel Using Hex Protocol

This example program demonstrates how to interface to an Optimate KM420 HMI panel using the Optimate's "Hex protocol". In this case the KM420 is a slave and the PLC is the master. Command data are stored in DM[3951] till DM[3950+X] and the "SendHex" function is called with the correct value of 'X' variable to send out the command and obtain the response. If comm error occurred an output with label name "CommError" will be turned on as an indicator that some trouble occurred.

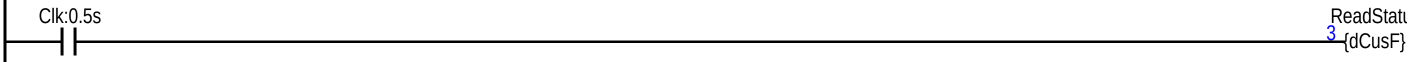
2



3

The PLC needs to talk to the HMI periodically to keep it happy and also to extract its general status register value (which gives the logic states of the 4 push buttons) and copy them to Relay #1 to #4 for use by the ladder program.

4

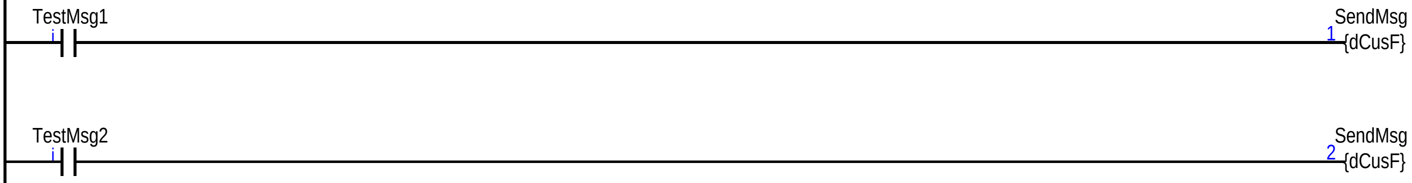


5

Turn on Input #1 and #2 ("TestMsg1" and "TestMsg2") to see the selected test Messages displayed on both the top and the bottom LCD lines.

6

7



8

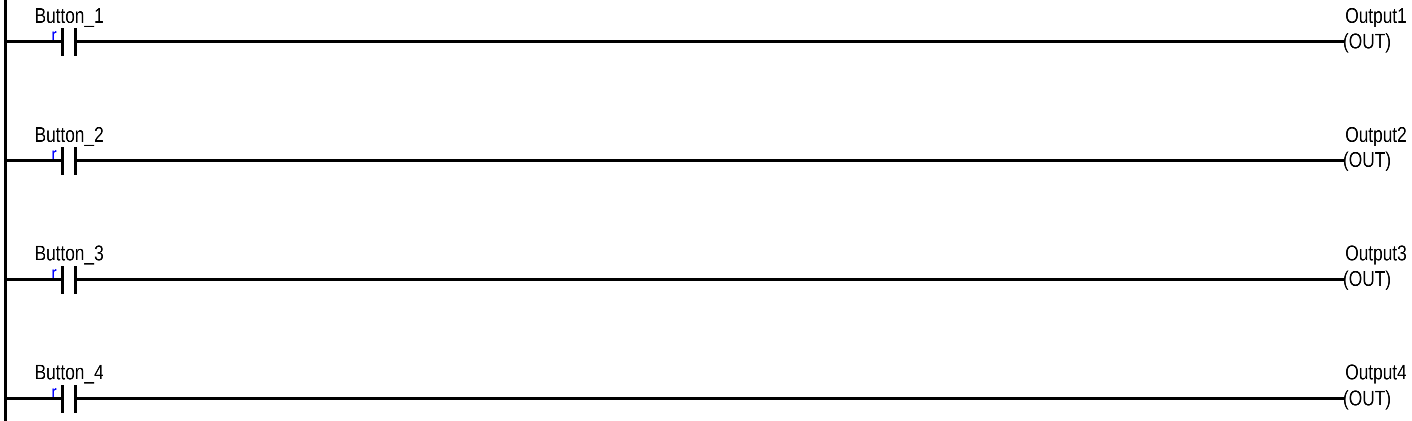
The following circuits use the logic states of the four push button on the KM420 panel . When you press every button on the panel, a corresponding output will be turned ON.

9

10

11

12



I/O #	Inputs	Outputs	Relays	Relay+256	Timers	S.V.	Counters	S.V.
1	TestMsg1	Output1	Button_1					
2	TestMsg2	Output2	Button_2					
3	DefaultMsg	Output3	Button_3					
4		Output4	Button_4					
5								
6								
7								
8		CommError						
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								
39								
40								
41								
42								
43								
44								

1 SendMsg1

```
DM[3953]= &HA1      ' Function A1 - select Predefined Message
DM[3954] = 0        ' select top line
DM[3955] = 1        ' select message #1
DM[3956] = 0        ' 4 bytes of data.
DM[3957] = 0
DM[3958] = &H55
DM[3959] = &HAA
X = 9                ' Total nine bytes.
call SendHex

DM[3954] = 1          ' select bottom line
DM[3955] = 6          ' select predefined message #6

call SendHex          ' Now display bottom line.
```

2 SendMsg2

```
DM[3953]= &HA1      ' Function A1 - select Predefined Message
DM[3954] = 0        ' select top line
DM[3955] = 2        ' select message #2
DM[3956] = 0        ' 4 bytes of data.
DM[3957] = 0
DM[3958] = &H55
DM[3959] = &HAA
X = 9                ' Total nine bytes.
call SendHex


DM[3954] = 1          ' select bottom line
DM[3955] = 7          ' select predefined message #7

call SendHex          ' Now display bottom line.
```

3 ReadStatus

```
' Display last selected messages to keep Optimate happy
' -------------------------------------------------------

DM[3953] = &HA1    ' Function A1 - Select Predefined Message
X = 9              ' Total nine bytes.
call SendHex       ' Send the last A1 function


' Query the status of the four push buttons.
'-------------------------------------------

DM[3953]= &HA0     ' Function A0 - General Status/Control
X = 3              ' Total 3 bytes.
call SendHex

IF DM[3991]= 2 AND DM[3992]=DM[3993]   ' receive an STX and checksum is correct
   RELAY[1] = DM[3992]   ' button status is copied to RELAY #1 to #4. But
ENDIF                    ' relay #5 to #16 are also used up here and hence they are
                         ' no longer useable by the ladder program
```

10 SendHex

```
' This function takes the data from DM[3951] to DM[3951+X]
' and send them out via COMM port 1. X indicates the number
' of bytes to be sent. At the end, a check sum is performed
' for DM[3953] to DM[3951+X] and sent out via COMM 1 as well.

FOR I = 1 TO 256
  IF INCOMM(1) < 0 GOTO @5  ' clear incomm buffer.
  ENDIF
NEXT

@5
C = 0               ' for checksum
FOR I = 3 to X
   C =C + DM[3950+I]
NEXT

FOR I = 1 to X         ' send out all the data
  OUTCOMM 1, DM[3950+I]
NEXT
OUTCOMM 1, C   ' send out the checksum

FOR I = 1 TO 200
   C = INCOMM(1)
   IF C >= 0 GOTO @10: ENDIF    ' start to receive data
NEXT

IF I >= 200        ' receive no response from Optimate.
   SETIO CommError  ' indicate error communicating with Optimate.
   RETURN
ENDIF


@10
FOR I = 0 TO 9
   call 20             ' store received characters for debugging purpose.
   DM[3991+I] = C
   C = INCOMM(1)       ' check if there are more characters
   IF C < 0            ' No more characters.
      IF DM[3991+I] = &H15    ' last data received is NAK
         SETIO CommError      ' indicate error communicating with Optimate.
      Else
         CLRIO CommError      ' should be a successful communication.
         RETURN
      ENDIF
   ENDIF
NEXT
```

20

```
' This routine store the received data in DM[1] to DM[256] which
' can be used for debugging purpose. This routine may be eliminated
' when test is completed.
' -----------------------------------

IF C < 0 return   ' if C <> -1 it contains the received data from INCOMM
ENDIF

J = J+1
IF J > 256
   J = 1
ENDIF

DM[J] = C   ' store the data.
```

100

```
SETBAUD 1,3        ' Set COMM1 to 9600 bps.
                   ' The command data for the Hex protocol are stored
                   ' in DM[3950] onwards.
DM[3951]= 2        ' STX character
DM[3952] = 0       ' module address
DM[3953]= &HA1     ' Function A1 - select Predefined Message
DM[3954] = 0       ' select top line
DM[3955] = 8       ' select message #8
DM[3956] = 0       ' 4 bytes of data.
DM[3957] = 0
DM[3958] = 0
DM[3959] = 0
X = 9               ' Total nine bytes.
call SendHex

DM[3954] = 1         ' select bottom line
DM[3955] = 9         ' select predefined message #9

call SendHex         ' Now display bottom line.
```