

PLEASE READ HV-INSTEON.doc BEFORE YOU ATTEMPT TO USE ANY OF THIS

(last updated: May 7, 2008)

MACRO EVENT # 10 'INSTEON-Turn Group On'

```
; All INSTEON commands begin with 0x02
Serial port 3 (INSTEON PLM): Transmit bytes '02'
; 0x61 = Send ALL-Link Command
Serial port 3 (INSTEON PLM): Transmit bytes '61'
; Send the Group Number
Serial port 3 (INSTEON PLM): Transmit variable #25 (INSTEON Group
Number) as 1 binary byte
; 0x11 = Command to turn on the group
; 0x00 = Indicates the initial broadcast of the command
; (the IM will use this last byte for cleanup messages)
Serial port 3 (INSTEON PLM): Transmit bytes '11 00'
```

MACRO EVENT # 11 'INSTEON-Turn Group Off'

```
; All INSTEON commands begin with 0x02
Serial port 3 (INSTEON PLM): Transmit bytes '02'
; 0x61 = Send ALL-Link Command
Serial port 3 (INSTEON PLM): Transmit bytes '61'
; Send the Group Number
Serial port 3 (INSTEON PLM): Transmit variable #25 (INSTEON Group
Number) as 1 binary byte
; 0x13 = Command to turn off the group
; 0x00 = Indicates the initial broadcast of the command
; (the IM will use this last byte for cleanup messages)
Serial port 3 (INSTEON PLM): Transmit bytes '13 00'
```

MACRO EVENT # 14 'INSTEON-Turn Device On'

```
    ; All INSTEON commands begin with 0x02
    Serial port 3 (INSTEON PLM): Transmit bytes '02'
    ; '0x62 indicates a standard or extended length command (which are
fixed lengths)
    Serial port 3 (INSTEON PLM): Transmit bytes '62'
    ; When sending a standard command to a specific device, send that
Device ID
    Do Macro #25 (INSTEON-Send Device ID) once
    ; Now send the command to turn on the device
    ; (FF is the brightness level)
    Serial port 3 (INSTEON PLM): Transmit bytes '05 11 FF'
```

MACRO EVENT # 18 'INSTEON-Turn Device Off'

```
    ; All INSTEON commands begin with 0x02
    Serial port 3 (INSTEON PLM): Transmit bytes '02'
    ; '0x62 indicates a standard or extended length command (which are
fixed lengths)
    Serial port 3 (INSTEON PLM): Transmit bytes '62'
    ; When sending a standard command to a specific device, send that
Device ID
    Do Macro #25 (INSTEON-Send Device ID) once
    ; Now send the command to turn on the device
    ; (00 is the brightness level)
    Serial port 3 (INSTEON PLM): Transmit bytes '05 11 00'
```

MACRO EVENT # 25 'INSTEON-Send Device ID'

```
    ; 1=Lamp Module for Testing
    If
        Var #24 (INSTEON Device Number) = 1
    Then
        Serial port 3 (INSTEON PLM): Transmit bytes '07 9D AF'
    End If
    ; 2=Main Room Dimmable Lights (A1)
    If
        Var #24 (INSTEON Device Number) = 2
    Then
        Serial port 3 (INSTEON PLM): Transmit bytes '07 18 D8'
    End If
```

<***REPEAT THE PATTERN ABOVE FOR EACH INSTEON DEVICE***>

MACRO EVENT # 16 'INSTEON-Get Version'

Serial port 3 (INSTEON PLM): Transmit bytes '02 60'

MACRO EVENT # 19 'INSTEON-Start ALL-Linking Grp'

; The 0x01 is a flag denoting that the PLM is a controller

Serial port 3 (INSTEON PLM): Transmit bytes '02 64 01'

; Send the Group Number

Serial port 3 (INSTEON PLM): Transmit variable #25 (INSTEON Group Number) as 1 binary byte

MACRO EVENT # 20 'INSTEON-Start Unlinking Grp'

; The 0xFF is a flag denoting unlinking mode

Serial port 3 (INSTEON PLM): Transmit bytes '02 64 FF'

; Send the Group Number

Serial port 3 (INSTEON PLM): Transmit variable #25 (INSTEON Group Number) as 1 binary byte

MACRO EVENT # 21 'INSTEON-Cancel ALL-Linking'

Serial port 3 (INSTEON PLM): Transmit bytes '02 65'

MACRO EVENT # 22 'INSTEON-Get First DB Entry'

Serial port 3 (INSTEON PLM): Transmit bytes '02 69'

MACRO EVENT # 23 'INSTEON-Get Next DB Entry'

Serial port 3 (INSTEON PLM): Transmit bytes '02 6A'

MACRO EVENT # 15 'INSTEON-Serial Receive'

```
    ; This macro assumes this is a valid INSTEON command as indicated by
0x02
    ; and checked within the Serial Data Input Event (before calling
this macro)
    ; This also assumes the variable for serial data length has been set
If
    Flag #5 (Forward All INSTEON Commands) is set
Then
    ; We really only need all commands being forwarded when debugging
    Do Macro #17 (INSTEON-Translate to Main Port) once
End If
Serial port 3 (INSTEON PLM): Put binary value of char 2 into Result
Value
Var #19 (Serial Data Current Byte) = Result Value
If
    ; Standard Message Received = 0x50 = 80
    Var #19 (Serial Data Current Byte) = 80
Then
    Serial port 1 (Main serial port): Transmit string 'INSTEON:
Standard Message (0x50) Received (From ID = '
    Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    Serial port 3 (INSTEON PLM): Put binary value of char 4 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    Serial port 3 (INSTEON PLM): Put binary value of char 5 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    Serial port 1 (Main serial port): Transmit string ', To ID = '
    Serial port 3 (INSTEON PLM): Put binary value of char 6 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    Serial port 3 (INSTEON PLM): Put binary value of char 7 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    Serial port 3 (INSTEON PLM): Put binary value of char 8 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
```

```

        Serial port 1 (Main serial port): Transmit string ', Command = '
        Serial port 3 (INSTEON PLM): Put binary value of char 10 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 1 (Main serial port): Transmit string ')'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
    End If
    If
        ; Standard Message Received = 0x53 = 83
        Var #19 (Serial Data Current Byte) = 83
    Then
        Serial port 1 (Main serial port): Transmit string 'INSTEON: ALL-
Linking Completed (0x53) (Link Code = '
        Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 3 (INSTEON PLM): Put binary value of char 4 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit string ', Group '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as ASCII without leading zeros
        Serial port 3 (INSTEON PLM): Put binary value of char 5 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit string ', ID = '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 3 (INSTEON PLM): Put binary value of char 6 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 3 (INSTEON PLM): Put binary value of char 7 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 1 (Main serial port): Transmit string ')'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
    End If
    If
        ; ALL-Link Cleanup Status Command = 0x58 = 88
        Var #19 (Serial Data Current Byte) = 88
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
        Var #23 (INSTEON ACK/NAK) = Result Value
    If

```

```

        ; 0x06=ACK, 0x13=NAK
        Var #23 (INSTEON ACK/NAK) = 6
    Then
        Serial port 1 (Main serial port): Transmit string 'INSTEON:
ALL-Link Cleanup Status (0x58) succeeded'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
    Else
        Serial port 1 (Main serial port): Transmit string 'INSTEON:
ALL-Link Cleanup Status (0x58) failed'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
    End If
End If
If
    ; Get IM Info = 0x60 = 96
    Var #19 (Serial Data Current Byte) = 96
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 9 into
Result Value
        Var #23 (INSTEON ACK/NAK) = Result Value
        If
            ; 0x06=ACK, 0x13=NAK
            Var #23 (INSTEON ACK/NAK) = 6
        Then
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Get IM Info (0x60) succeeded (IM ID = '
            Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
            Var #19 (Serial Data Current Byte) = Result Value
            Var #20 (INSTEON ID high byte) = Var #19 (Serial Data Current
Byte)
            Serial port 1 (Main serial port): Transmit variable #19
(Serial Data Current Byte) as 2 hex bytes
            Serial port 3 (INSTEON PLM): Put binary value of char 4 into
Result Value
            Var #19 (Serial Data Current Byte) = Result Value
            Var #21 (INSTEON ID middle byte) = Var #19 (Serial Data
Current Byte)
            Serial port 1 (Main serial port): Transmit string ' '
            Serial port 1 (Main serial port): Transmit variable #19
(Serial Data Current Byte) as 2 hex bytes
            Serial port 3 (INSTEON PLM): Put binary value of char 5 into
Result Value
            Var #19 (Serial Data Current Byte) = Result Value
            Var #22 (INSTEON ID low byte) = Var #19 (Serial Data Current
Byte)
            Serial port 1 (Main serial port): Transmit string ' '
            Serial port 1 (Main serial port): Transmit variable #19
(Serial Data Current Byte) as 2 hex bytes
            Serial port 1 (Main serial port): Transmit string ')'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        Else
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Get IM Info failed'

```

```

        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        End If
    End If
    If
        ; ALL-Link Command = 0x61 = 97
        Var #19 (Serial Data Current Byte) = 97
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 6 into
Result Value
        Var #23 (INSTEON ACK/NAK) = Result Value
        If
            ; 0x06=ACK, 0x13=NAK
            Var #23 (INSTEON ACK/NAK) = 6
        Then
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Send ALL-Link Command (0x61) succeeded (Group '
            Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
            Var #19 (Serial Data Current Byte) = Result Value
            Serial port 1 (Main serial port): Transmit variable #19
            (Serial Data Current Byte) as ASCII without leading zeros
            Serial port 3 (INSTEON PLM): Put binary value of char 4 into
Result Value
            Var #19 (Serial Data Current Byte) = Result Value
            Serial port 1 (Main serial port): Transmit string ', Command '
            Serial port 1 (Main serial port): Transmit variable #19
            (Serial Data Current Byte) as 2 hex bytes
            Serial port 3 (INSTEON PLM): Put binary value of char 5 into
Result Value
            Var #19 (Serial Data Current Byte) = Result Value
            Serial port 1 (Main serial port): Transmit string ' '
            Serial port 1 (Main serial port): Transmit variable #19
            (Serial Data Current Byte) as 2 hex bytes
            Serial port 1 (Main serial port): Transmit string ')'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        Else
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Send ALL-Link Command (0x61) failed'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        End If
    End If
    If
        ; Send Standard-length Message = 0x62 = 98
        Var #19 (Serial Data Current Byte) = 98
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 9 into
Result Value
        Var #23 (INSTEON ACK/NAK) = Result Value
        If
            ; 0x06=ACK, 0x13=NAK
            Var #23 (INSTEON ACK/NAK) = 6
        Then
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Send Standard-length Message (0x62) succeeded (To ID = '

```

```

        Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Var #20 (INSTEON ID high byte) = Var #19 (Serial Data Current
Byte)
        Serial port 1 (Main serial port): Transmit variable #19
(Serial Data Current Byte) as 2 hex bytes
        Serial port 3 (INSTEON PLM): Put binary value of char 4 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Var #21 (INSTEON ID middle byte) = Var #19 (Serial Data
Current Byte)
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19
(Serial Data Current Byte) as 2 hex bytes
        Serial port 3 (INSTEON PLM): Put binary value of char 5 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Var #22 (INSTEON ID low byte) = Var #19 (Serial Data Current
Byte)
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19
(Serial Data Current Byte) as 2 hex bytes
        Serial port 1 (Main serial port): Transmit string ')'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        Else
        Serial port 1 (Main serial port): Transmit string 'INSTEON:
Send Standard-length Message (0x62) failed'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        End If
    End If
    If
        ; Start ALL-Linking Command = 0x64 = 100
        Var #19 (Serial Data Current Byte) = 100
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 5 into
Result Value
        Var #23 (INSTEON ACK/NAK) = Result Value
        If
            ; 0x06=ACK, 0x13=NAK
            Var #23 (INSTEON ACK/NAK) = 6
        Then
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Start ALL-Link Command (0x64) succeeded (Group '
            Serial port 3 (INSTEON PLM): Put binary value of char 4 into
Result Value
            Var #19 (Serial Data Current Byte) = Result Value
            Serial port 1 (Main serial port): Transmit variable #19
(Serial Data Current Byte) as ASCII without leading zeros
            Serial port 1 (Main serial port): Transmit string ')'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        Else
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Start ALL-Linking Command (0x64) failed'

```

```

        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        End If
    End If
    If
        ; Cancel ALL-Linking Command = 0x65 = 101
        Var #19 (Serial Data Current Byte) = 101
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
        Var #23 (INSTEON ACK/NAK) = Result Value
        If
            ; 0x06=ACK, 0x13=NAK
            Var #23 (INSTEON ACK/NAK) = 6
        Then
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Cancel ALL-Linking (0x65) succeeded'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        Else
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Cancel ALL-Linking (0x65) failed'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        End If
    End If
    If
        ; Get First Database Entry Command = 0x69 = 105
        Var #19 (Serial Data Current Byte) = 105
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
        Var #23 (INSTEON ACK/NAK) = Result Value
        If
            ; 0x06=ACK, 0x13=NAK
            Var #23 (INSTEON ACK/NAK) = 6
        Then
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Get First Database Entry (0x69) succeeded'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
            ; Check for an ALL-Link Record Response that immediately
followed the ACK
            If
                ; HV always adds an extra byte (0x0D) so see if we have
more beyond that
                Var #18 (Serial Data Length) > 4
            Then
                Do Macro #24 (INSTEON-Receive Serial Byte 4) once
            End If
        Else
            Serial port 1 (Main serial port): Transmit string 'INSTEON:
Get First Database Entry (0x69) failed'
            Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        End If
    End If

```

```

If
    ; Get Next Database Entry Command = 0x6A = 106
    Var #19 (Serial Data Current Byte) = 106
Then
    Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
    Var #23 (INSTEON ACK/NAK) = Result Value
    If
        ; 0x06=ACK, 0x13=NAK
        Var #23 (INSTEON ACK/NAK) = 6
    Then
        Serial port 1 (Main serial port): Transmit string 'INSTEON:
Get Next Database Entry (0x6A) succeeded'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
        ; Check for an ALL-Link Record Response that immediately
followed the ACK
        If
            ; HV always adds an extra byte (0x0D) so see if we have
more beyond that
            Var #18 (Serial Data Length) > 4
        Then
            Do Macro #24 (INSTEON-Receive Serial Byte 4) once
        End If
    Else
        Serial port 1 (Main serial port): Transmit string 'INSTEON:
Get Next Database Entry (0x6A) failed'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
    End If
End If

```

```

MACRO EVENT # 17 'INSTEON-Translate to Main Port'

    Serial port 3 (INSTEON PLM): Put binary value of char 2 into Result
Value
    Var #19 (Serial Data Current Byte) = Result Value
    If
        Var #19 (Serial Data Current Byte) = 82
    Then
        Serial port 1 (Main serial port): Transmit string 'INSTEON X10
Received:'
    Else
        Serial port 1 (Main serial port): Transmit string 'INSTEON:'
    End If
    ; Now just pass along each byte, formatted in hex so that it is
easily matched up
    ; with the INSTEON documentation (which gives everything in hex)
    Serial port 3 (INSTEON PLM): Put binary value of char 1 into Result
Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial Data
Current Byte) as 2 hex bytes
    If
        Var #18 (Serial Data Length) >= 2
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 2 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    End If
    If
        Var #18 (Serial Data Length) >= 3
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 3 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    End If
    If
        Var #18 (Serial Data Length) >= 4
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 4 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
    End If
    If
        Var #18 (Serial Data Length) >= 5
    Then
        Serial port 3 (INSTEON PLM): Put binary value of char 5 into
Result Value

```

```

    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
    Var #18 (Serial Data Length) >= 6
Then
    Serial port 3 (INSTEON PLM): Put binary value of char 6 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
    Var #18 (Serial Data Length) >= 7
Then
    Serial port 3 (INSTEON PLM): Put binary value of char 7 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
    Var #18 (Serial Data Length) >= 8
Then
    Serial port 3 (INSTEON PLM): Put binary value of char 8 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
    Var #18 (Serial Data Length) >= 9
Then
    Serial port 3 (INSTEON PLM): Put binary value of char 9 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
    Var #18 (Serial Data Length) >= 10
Then
    Serial port 3 (INSTEON PLM): Put binary value of char 10 into
Result Value
    Var #19 (Serial Data Current Byte) = Result Value
    Serial port 1 (Main serial port): Transmit string ' '
    Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
    Var #18 (Serial Data Length) >= 11

```

```

Then
  Serial port 3 (INSTEON PLM): Put binary value of char 11 into
Result Value
  Var #19 (Serial Data Current Byte) = Result Value
  Serial port 1 (Main serial port): Transmit string ' '
  Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
  Var #18 (Serial Data Length) >= 12
Then
  Serial port 3 (INSTEON PLM): Put binary value of char 12 into
Result Value
  Var #19 (Serial Data Current Byte) = Result Value
  Serial port 1 (Main serial port): Transmit string ' '
  Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
  Var #18 (Serial Data Length) >= 13
Then
  Serial port 3 (INSTEON PLM): Put binary value of char 13 into
Result Value
  Var #19 (Serial Data Current Byte) = Result Value
  Serial port 1 (Main serial port): Transmit string ' '
  Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
  Var #18 (Serial Data Length) >= 14
Then
  Serial port 3 (INSTEON PLM): Put binary value of char 14 into
Result Value
  Var #19 (Serial Data Current Byte) = Result Value
  Serial port 1 (Main serial port): Transmit string ' '
  Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
  Var #18 (Serial Data Length) >= 15
Then
  Serial port 3 (INSTEON PLM): Put binary value of char 15 into
Result Value
  Var #19 (Serial Data Current Byte) = Result Value
  Serial port 1 (Main serial port): Transmit string ' '
  Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
End If
If
  Var #18 (Serial Data Length) >= 16
Then
  Serial port 3 (INSTEON PLM): Put binary value of char 16 into
Result Value
  Var #19 (Serial Data Current Byte) = Result Value
  Serial port 1 (Main serial port): Transmit string ' '
  Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes

```

```
End If
Serial port 1 (Main serial port): Transmit string ' ('
Serial port 1 (Main serial port): Transmit variable #18 (Serial Data
Length) as ASCII without leading zeros
Serial port 1 (Main serial port): Transmit string ' bytes)'
Serial port 1 (Main serial port): Transmit Carriage return and Line
feed
```

MACRO EVENT # 24 'INSTEON-Receive Serial Byte 4'

```
    Serial port 3 (INSTEON PLM): Put binary value of char 4 into Result
Value
    Var #19 (Serial Data Current Byte) = Result Value
    ; If this isn't the start of a new INSTEON command, just get out
    If
        Var #19 (Serial Data Current Byte) <> 2
    Then
        Controller command: Exit from all actions now
    End If
    Serial port 3 (INSTEON PLM): Put binary value of char 5 into Result
Value
    Var #19 (Serial Data Current Byte) = Result Value
    If
        ; ALL-Link Record Response = 0x57 (87)
        Var #19 (Serial Data Current Byte) = 87
    Then
        Serial port 1 (Main serial port): Transmit string 'INSTEON: ALL-
Link Record Response (0x57) (Group '
        Serial port 3 (INSTEON PLM): Put binary value of char 7 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as ASCII without leading zeros
        Serial port 1 (Main serial port): Transmit string ', INSTEON ID =
,
        Serial port 3 (INSTEON PLM): Put binary value of char 8 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Var #20 (INSTEON ID high byte) = Var #19 (Serial Data Current
Byte)
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 3 (INSTEON PLM): Put binary value of char 9 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Var #21 (INSTEON ID middle byte) = Var #19 (Serial Data Current
Byte)
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 3 (INSTEON PLM): Put binary value of char 10 into
Result Value
        Var #19 (Serial Data Current Byte) = Result Value
        Var #22 (INSTEON ID low byte) = Var #19 (Serial Data Current
Byte)
        Serial port 1 (Main serial port): Transmit string ' '
        Serial port 1 (Main serial port): Transmit variable #19 (Serial
Data Current Byte) as 2 hex bytes
        Serial port 1 (Main serial port): Transmit string ')'
        Serial port 1 (Main serial port): Transmit Carriage return and
Line feed
    End If
```

MACRO EVENT # 26 'All Main Lights On'

```
If
  Flag #6 (Prefer INSTEON over X10) is set
Then
  Var #25 (INSTEON Group Number) = 2
  Do Macro #10 (INSTEON-Turn Group On) once
Else
  X-10: Transmit House Code 'A' All Lights On
End If
```

MACRO EVENT # 27 'All Main Lights Off'

```
If
  Flag #6 (Prefer INSTEON over X10) is set
Then
  Var #25 (INSTEON Group Number) = 2
  Do Macro #11 (INSTEON-Turn Group Off) once
Else
  X-10: Transmit House Code 'A' All Units Off
End If
```