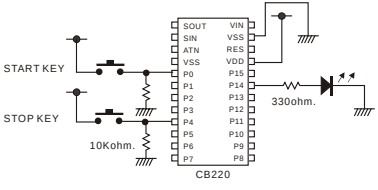
Application Notes

NOTE 1. Switch Input

Let's say for example you are developing some kind of a machine, the first thing you need is a user interface. Our task today is to build a machine that will receive input from a switch and processes it to its assigned task..

We will make a START and STOP button that will light a lamp ON and OFF.

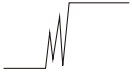


As you can see above, P0 and P4 ports will be connected to a pull-down resistor (resistor attached to ground). CB220 will read these switches as LOW or OFF when the switch is not pressed. To find out if these switches are pressed or unpressed, we can use CUBLOC BASIC command IN().
<Filename: startstopkey.cul>

```
Const Device = cb220

Dim a As Byte
Do
      If In(0) = 1 Then a = 1
      If In(4) = 1 Then a = 0
      Out 14,a
Loop
```

When the switch is pressed, a “bouncing” effect occurs from the switch’s mechanical spring.



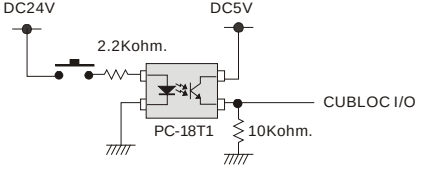
The above picture shows how bouncing can confuse CUBLOC controller by bouncing up and down. To get rid of this bouncing effect, a capacitor and resistor can be added to filter it out.
A simpler method is to use the command KEYINH() rather than IN() which will remove the bouncing effect by software.
<Filename: keyinhinput.cul>

```
Const Device = cb220

Dim a As Byte
Do
      If Keyinh(0,20) = 1 Then a = 1
      If Keyinh(4,20) = 1 Then a = 0
      Out 14,a
Loop
```

The 2nd parameter of KEYINH(0, 20) sets the time for removing the bouncing effect, also called debouncing time. In other words, the 20 means to wait 20ms before accepting input.

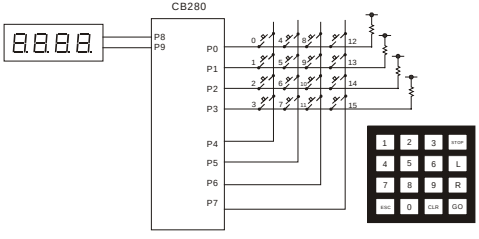
For the industrial field, there can be a lot of noisy environments where it can affect the switch signals. In order to block noise, the user can implement a circuit diagram similar to one shown below. By using a photocoupler, the user is able to raise the voltage and minimize the noise from affecting the switch.



<END>

NOTE 2. Keypad Input

Application note 2 will cover a 4 by 4 Keypad by taking its input and outputting the results to a 4 digit 7 segment module (CSG module)



The CSG module is a 4 digit seven segment LED module that can be connected via CUNET or I2C protocol to display numbers and custom characters.

<Filename: csgprint.cul>

```
Const Device = CB280
Set I2c 9,8
Dim I As Byte
Do
      Csgdec 0,I
      I = I + 1
Loop
```


If you connect CUNET to CSG and execute the above program, the CSG module will show numbers that will count up.

The key matrix can be read easily through the command KEYPAD. If you look carefully at the keypad, you will see that scancode does not match the actual key pressed. In order to read the correct key, we will use a KEYTABLE before outputting the value to the CSG.

<Filename: keypadnum.cul>

```
Const Device = CB280
Set I2c 9,8
Dim I As Integer
Dim K As Integer

Const Byte KEYTABLE = (1,4,7,10,2,5,8,0,3,6,9,11,12,13,14,15)
Do
      I=Keypad(0)
      If I < 16 Then
            I = KEYTABLE(I)
            Csgdec 0,I
      End If
Loop
```

And now, we will make a simple program that receives input. When a number key input is received, it is displayed to the CSG module as a 4 digit number. The number is stored into the variable K, which is in BCD code. We then use the function BCD2BIN to convert the BCD value back into binary.

<Filename: num4in.cul>

```
Const Device = CB280
Set I2c 9,8
Dim I As Integer
Dim K As Integer
Dim M As Integer
K = 0
Const Byte KEYTABLE = (1,4,7,10,2,5,8,0,3,6,9,11,12,13,14,15)
Do
      I=Keypad(0)
      If I < 16 Then
            I = KEYTABLE(I)
```

```
                If I < 10 Then
                      K = K << 4
                      K = K + I
                      Csghex 0,K
                End If
                '
                '     WAIT UNTIL KEY DEPRESS
                '
                Do While Keypad(0) < 255
                Loop
                M = Bcd2bin(K)
                Debug Dec M,CR
          End If
    Loop
```

When there is no input, the returned scancode is 255. By using Do While keypad(0) < 255, we will wait until a key is unpressed which will return a scancode of 255. This is to let the processor stop reading input while a key is pressed. Otherwise, the processor might receive multiple key inputs since execution time of CUBLOC is very fast.

By using _D(0) = M, you can pass the scancode value to relay D0 of LADDER LOGIC. If you need to use a keypad in LADDER, you can modify this code a little bit to get your results quick.

<END>

NOTE 3. Temperature Sensor

In our world today, there are countless number of devices that senses temperature. Refrigerator, heater, air conditioner, automobiles, and many other devices that uses temperature sensors. Therefore, this is one of the very basic components we must know.

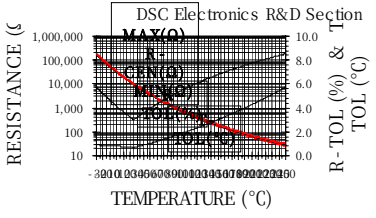
What types of temperature sensors are there? There is PT100, NTC, PTC thermistor, and other chip-type sensors such as the DS1620.

Today, we will dive into the NTC thermistor and figure out how to connect and use it with CUBLOC.

The NTC thermistor can be comparable to a very sensitive resistor. Depending on the temperature, the value of resistance will change. By reading the value of this resistance, we can figure out the current temperature. Among NTC thermistors, the ceramic types can sense around -20 to 130 degrees Celcius temperature.

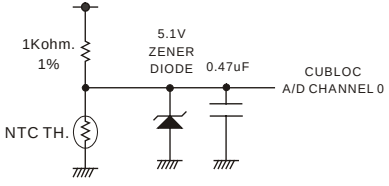
There is an NTC thermistor that resembles a diode. With this thermistor, we can sense between -30 and 250 degrees Celcius temperature.

You can acquire R-T(Resistance – Temperature) conversion table from the maker of the thermistor. The following is a diode-type 10K Ohm NTC Thermistor R-T conversion chart and table.



Temperature	Minimum	Average	Maximum
0	31260.0	32610.0	33987.7
1	29725.7	30993.7	32286.7
2	28275.6	29466.8	30680.6
3	26904.5	28023.9	29163.6
4	25607.8	26660.0	27730.3
5	24381.0	25370.2	26375.7
6	23220.0	24150.1	25094.9
7	22120.9	22995.7	23883.7
8	21080.1	21903.1	22737.7
9	20094.1	20868.5	21653.3
10	19159.9	19888.7	20626.7
11	18274.4	18960.5	19654.6
12	17434.8	18080.8	18733.8
13	16638.5	17246.9	17861.4
14	15883.1	16456.1	17034.4
15	15166.2	15706.0	16250.4
16	14485.7	14994.4	15506.9
17	13839.6	14318.9	14801.5
18	13225.9	13677.7	14132.2
19	12642.8	13068.7	13496.9
20	12088.7	12490.3	12893.6
21	11561.9	11940.6	12320.7
22	11061.0	11418.2	11776.4
23	10584.6	10921.6	11259.2
24	10131.3	10449.3	10767.5
25	9700.0	**10000.0**	10300.0
26	9281.3	9572.5	9864.0

For connecting the sensor to the CUBLOC, please refer to the following circuit diagram. To protect against voltage surges, the Zener diode must be used.



As you can see in the circuit diagram, we will be using A/D (Analog-to-Digital) converter to read the current voltage flowing through the sensor. The A/D converter will convert the current voltage into a value between 0 and 1024.

The most important part of this application note is the following table which converts the value of voltage to A/D value between 0 and 1024. (Only some of the temperatures are shown.)

Temp	Resistance	Voltage	A/D value
-30	175996.6	4.971750865	1018
-29	165473.9	4.969965259	1018
-28	155643.6	4.968080404	1017
-27	146456.3	4.966091647	1017
-26	137866.4	4.963994167	1017
-25	129831.7	4.961782976	1016
-24	122313.4	4.959452909	1016
-23	115275.4	4.956998627	1015
-22	108684.3	4.954414614	1015
-21	102509.3	4.951695171	1014
-9	52288.3	4.90617073	1005
-8	49549.7	4.901087406	1004
-7	46970.5	4.895769279	1003
-6	44540.6	4.890207868	1002
-5	42250.5	4.884394522	1000
-4	40091.5	4.878320427	999
-3	38055.4	4.871976604	998
-2	36134.4	4.865353924	996
-1	34321.5	4.858443112	995
0	32610.0	4.851234752	994
1	30993.7	4.8437193	992
2	29466.8	4.835887094	990
3	28023.9	4.827728362	989

4	26660.0	4.819233234	987
5	25370.2	4.810391755	985
6	24150.1	4.801193902	983
7	22995.7	4.79162959	981
8	21903.1	4.781688696	979
9	20868.5	4.771361072	977
10	19888.7	4.760636561	975
11	18960.5	4.749505017	973
12	18080.8	4.737956327	970
13	17246.9	4.725980424	968
14	16456.1	4.713567319	965
15	15706.0	4.700707114	963
16	14994.4	4.68739003	960
17	14318.9	4.673606431	957
18	13677.7	4.659346849	954
19	13068.7	4.644602011	951
20	12490.3	4.629362861	948
21	11940.6	4.613620595	945
22	11418.2	4.597366683	942
23	10921.6	4.580592903	938
24	10449.3	4.563291365	935
25	10000.0	4.545454545	931
26	9572.5	4.527075313	927
27	9165.6	4.508146954	923
28	8778.3	4.488663246	919
29	8409.4	4.468618396	915
30	8058.1	4.448007162	911
31	7723.3	4.426824842	907
32	7404.3	4.405067304	902
33	7100.2	4.382731022	898
34	6810.2	4.359813102	893
35	6533.7	4.336311306	888
36	6269.8	4.312224084	883
37	6018.0	4.287550592	878
38	5777.7	4.262290722	873
39	5548.3	4.236445118	868
50	3606.1	3.914475937	802
51	3472.1	3.881948015	795
52	3343.7	3.848917708	788
53	3220.8	3.815397329	781
54	3103.1	3.781399998	774
55	2990.2	3.746939622	767
56	2882.1	3.712030877	760
57	2778.4	3.676689176	753
58	2679.0	3.640930651	746
59	2583.6	3.604772114	738
81	1220.4	2.748157207	563
82	1181.9	2.7084025	555
83	1144.8	2.668747011	547
84	1109.0	2.629210536	538
85	1074.5	2.589812422	530

86	1041.3	2.550571543	522
87	1009.2	2.511506263	514
88	978.3	2.472634416	506
89	948.5	2.433973277	498
90	919.8	2.395539544	491
91	892.0	2.357349316	483
92	865.3	2.319418079	475
93	839.4	2.281760687	467
94	814.5	2.244391354	460
95	790.4	2.207323646	452
96	767.1	2.170570465	445
97	744.7	2.134144055	437
98	723.0	2.098055989	430
99	702.0	2.062317177	422
100	681.8	2.026937858	415
101	662.2	1.99192761	408
102	643.3	1.957295352	401
103	625.0	1.92304935	394
104	607.3	1.889197225	387
105	590.2	1.855745964	380
106	573.7	1.822701928	373
107	557.7	1.790070865	367
108	542.2	1.757857926	360
109	527.2	1.726067674	353
239	33.5	0.162295782	93
240	33.0	0.159800146	93
241	32.5	0.157350769	92
242	32.0	0.154946682	92
243	31.5	0.152586936	91
244	31.0	0.150270604	91
245	30.5	0.147996779	90
246	30.0	0.145764577	90
247	29.6	0.143573131	89
248	29.1	0.141421596	89
249	28.7	0.139309144	89
250	28.2	0.137234968	88

```
      '
      '     NTC THERMISTOR READ TABLE
      '     10K DIODE TYPE
      '
      Const Device = cb280

      Const             Integer             TH_TABLE             =
(992,990,989,987,985,983,981,979,977,975,
                  973,970,968,965,963,960,957,954,951,948,
                  945,942,938,935,931,927,923,919,915,911,
                  907,902,898,893,888,883,878,873,868,862,
                  857,851,845,839,833,827,821,815,808,802,
```

```
                795,788,781,774,767,760,753,746,738,731,
                723,716,708,700,692,684,677,669,661,652,
                644,636,628,620,612,604,596,587,579,571,
                563,555,547,538,530,522,514,506,498,491,
                483,475,467,460,452,445,437,430,422,415)

Dim a As Integer,b As Integer
Do
      b = Tadin(0)
      If b > 990 Or b < 400 Then
            Debug "Out of Range"    'Check short or open th.
      End If
      For a=0 To 100
            If b > TH_TABLE(a) Then Exit For
      Next
      Debug Dec a,cr
      Delay 500
Loop
```

<Filename: ntcth.cul>

By using the TADIN command for AD conversion, CUBLOC will automatically calculate the average of 10 A/D conversion reads. By using this command, you get more precise results. The sample program shown here will be able to sense between 0 to 100 degrees. For larger range, you can simply modify the code.

The formula for acquiring A/D conversion value from the R-T table is as follows:

$$V = \frac{5}{(1000 + THR)} \times THR$$

THR is the resistance value. 1000 is for 1K Ohm resistor and 5 is for 5 volts. The 10 bit A/D converter of CUBLOC will return a value between 0 and 1024. There for to get the A/D value, you must multiply result V by 204.8. You can easily make a chart by using an excel spreadsheet to enter these formulas. <END>

NOTE 4.
Connect to the Internet through XPORT

In this application note, we will explain how to connect to the internet using XPORT internet module. By using an XPORT, you can download and monitor your programs through the internet.
For applications that need customer service and updates, you can use XPORT.

By using XPORT, you will be able to check the status of your machine from California to New York and download new programs to your CUBLOC module by using our Java applications. We provide open-source Java applications in which you can simply edit to customize to your project.

No special coding is necessary for the basic monitoring and downloading. Simply connect the XPORT to CUBLOC.
You can use XPORT Dongle, which has MAX232 chip to convert RS232 signal from 3.3V to 12V. This XPORT Dongle is customized to be used with CUBLOC Study Board, CuTOUCH, proto-boards, and baseboards by connecting 5V to pin 9 of RS232 connectors. To use dongle elsewhere, you will have to input 5V to pin 9 of the RS232 connector being used.

Please check out www.comfiletech.com forum for XPORT applications, downloads and detailed information.

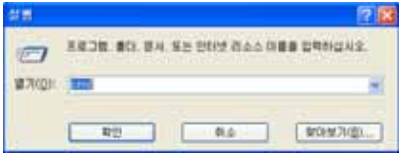
To use XPORT, you will need to get XPORT DeviceInstaller to set the XPORT for the first time.

When Search button is pressed, XPORTs connected to your network will show up. Please record the IP address.

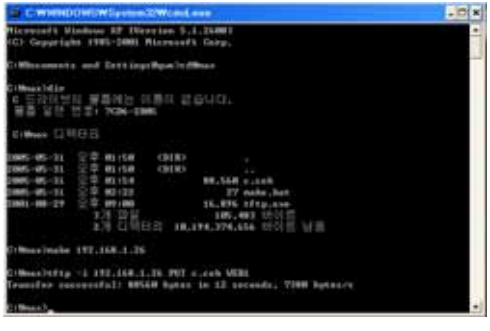
After selecting the XPORT to configure, please click on Port ->Setup and setup the parameters as shown above.

After setting up the XPORT, you can install a Java applet to the XPORT to enable monitoring and downloading through the internet.

Please go to Run and type cmd to go to DOS command line as shown below:



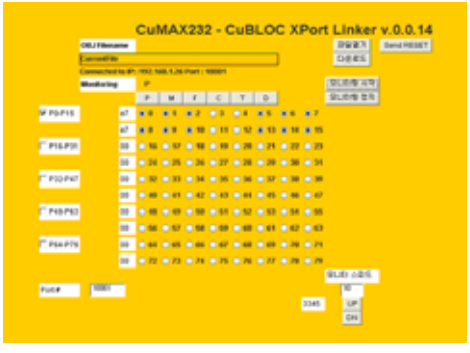
Please type as shown below, “make *IP address*”.



The IP address here is the IP address you have recorded earlier with the DeviceInstaller.

Please make sure you have java software installed on your computer by going to www.java.com.

And now for the final part, simply type the IP address on the Internet Explorer.
Please click “Yes” on the certificate window.



You can click on “Start Monitor” to see the monitoring screen. P, M, F, and other relay status can be seen in real-time.

Select “Open File”, select an CUBLOC object file, and press “Download’.

And now, you have the ability to upgrade your CUBLOC module without actually being there.

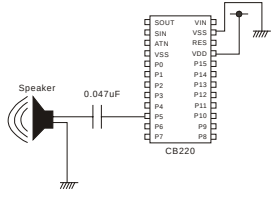
For more detailed information and updates, please check out CUBLOC forum at http://cubloc.com/phpBB2

Please refer to **Max’s XPORT Applications** our forum.

<END>

NOTE 5. Sound Bytes

In this application note, I will be showing you simple ways to create key touch sound, musical notes, and alert sound. An I/O port or a PWM Channel of CUBLOC can be used for sound. With a PWM Channel, you have the advantage of creating different tones of sounds.



The above example shows PWM Channel 0 of CB220 being used with Freqout command to produce a sound.

```
Const Device = cb280

Dim PLAYSTR As String
Low 5
Freqout 0,5236      'Create a sound with frequency of 440Hz
Delay 500           'Delay
Pwmoff 0            'Stop Sound by turning off PWM
```

With commands like Freqout and Delay, simple sounds can be created.

<Filename: playcdec.cul>

```
      Const Device = CB280
      Low 5
      Freqout 0,4403
      Delay 200
      Freqout 0,3703
      Delay 200
      Freqout 0,3114
      Delay 200
      Freqout 0,2202
      Delay 200
      Pwmoff 0
```

By changing frequencies, we have made a simple program that can play musical notes.

Octave 4							Octave 5						
A	B	C	D	E	F	G	A	B	C	D	E	F	G
A	B	C	D	E	F	G	H	I	J	K	L	M	N

To express one note, you can use 2 characters. The first character is for the note and second character is for the length of the note.

<Filename: play.cul>

```
      Const Device = cb280

      Dim PLAYSTR As String
      Low 5
      PLAYSTR = "G5E3E3G3E3C5"
      PLAY 0,PLAYSTR

      Do
      Loop
      End

Sub PLAY(CH As Byte,NOTE As String)
      Dim PL As Byte
      Dim CHAR As Byte
     Const Integer PLAYTABLE = (5236,4665,4403,3923,3495,3299,2939,
```



```
           2618,2333,2202,1961,1747,1649,1469,0)
      For PL=1 To Len(NOTE) Step 2
            CHAR = Asc(Mid(NOTE,PL,1)) - &H41
            Freqout CH,PLAYTABLE(CHAR)
            CHAR = Asc(Mid(NOTE,PL+1,1)) - &H30
            Delay CHAR*100
      Next
      Pwmoff CH
End Sub
```

When using PWM port for other purposes, Freqout command no longer becomes available for use. In this case, we can use any regular I/O port to create sound.
We will use TOGGLE and UDELAY commands to set the I/O pin to HIGH and LOW.
The following example shows how to make an alert sound with a regular I/O port, P4.
<Filename: playport.cul>

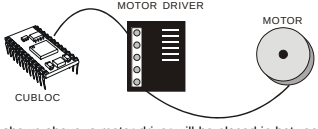
```
      Const Device = CB280
      Low 4
      Do
      SOUND 4,110,60
      SOUND 4,80,60
      SOUND 4,40,160
      Loop
      End

Sub SOUND(PN As Byte,FR As Byte,LN As Byte)
      Dim SI As Byte,SJ As Byte
      For SJ = 0 To LN
            Reverse PN
            Udelay FR
            Reverse PN
            Udelay FR
      Next
End Sub
```

<END>

NOTE 6.
Step Motor Pulse Generation

To enable a step motor, we will create a simple program that outputs pulses to the motor driver.



Like the picture shown above, a motor driver will be placed in between CUBLOC and the motor. When the motor driver receives pulses from CUBLOC, it will turn the MOTOR by 1.8 degrees for every pulse. This is not the case with all motor drivers, but you can apply this type of motor pulse generation to other applications.
<Filename: stepout.cul>

```
      Const Device = CB280
      Low 4
      Do
      STEPOUT 4,2,1000
      Loop
      End

Sub STEPOUT(PN As Byte,FR As Integer,LN As Long)
      Dim SJ As Long
      For SJ = 0 To LN
            Reverse PN
            Udelay FR
            Reverse PN
            Udelay FR
      Next
End Sub
```

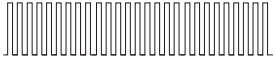
Here we will explain how to use the STEPOUT sub function. STEPOUT has 3 parameters PN, FR, and LN. The PN is for the port number used. Please make

sure to use an output port. The second parameter FR is the length of the pulse. The last parameter LN is for the number of pulses to send.

PN	PORT Number
FR	Pulse Length (0~65535)
LN	Number of Pulses (0~2147483647)

```
STEPOUT 2, 50, 9     ' Generate 9 pulses with length of 50
```

```
STEPOUT 2, 20, 30    ' Generate 30 pulses with length of 20
```



The pulse length is only a numerical value. The below table show its conversion to Frequency in Hz.

Pulse Length	Frequency (Hz)
1	2475
2	2381
10	1786
50	800
100	472
1000	57

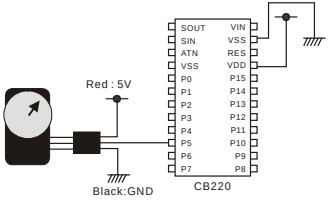
With this method, you can generate up to about 2475 pulses per second. For bigger frequencies you will need to either use the PWM or Freqout command. But you cannot control number of pulses with PWM or Freqout.

```
      Low 5
      Freqout 0,2  'Output pulses with frequency of 768 KHz
      Delay 500          'delay about 500ms
      Pwmoff 0           'Stop pulses
```

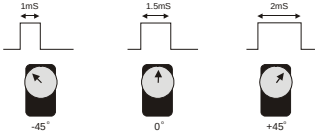
NOTE 7. RC Servo Motor

RC Servo Motors are used by many hobbyist to make remote control cars, planes, and etc… In the recent years, it has been used for robot arms, legs, and other body parts.
With CUBLOC, you can use the PWM to easily implement an RC Servo motor into your project.

There are 3 wires to the RC servo motor. The black wire is ground and red wire is for power. The other yellow wire is for inputting PWM signal. For PWM signal, you can input about 60 pulses per second to enable the RC servo.



The RC Servo motor will move to a location set by pulse and duty value and will hold its position. By being able to control the exact angles at which the RC servo stops, we can control the RC servo as freely as we want.



A pulse of 1ms will stop the RC servo at -45 Degrees.
A pulse of 1.5ms will stop the RC servo at 0 Degrees.
A pulse of 2ms will stop the RC servo at +45 Degrees.
Depending on the RC servo you use, these specification will vary.

<Filename: rcservo.cul>

```
Const Device = CB280
Low 5
Pwm 0,2500,32768
```

When the code above is executed, a 1ms pulse will be outputted from port number 5. RC servo will position itself to -45 degrees.

```
Const Device = CB280
Low 5
Pwm 0,4000,32768
```

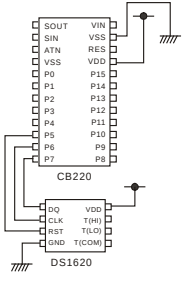
When the code above is executed, a 1.5ms pulse will be outputted from port number 5. RC servo will position itself to +45 degrees.

As you can see, by simply change the duty value of PWM command, RC servo can easily be controlled. For the CB220, 3 RC servos can be controlled simultaneously while the CB280 and CB290 can control 6 RC servos.

Warning: When the RC servo is in operation, it will need about 500mA of current, please make sure to use a power supply of at least 500mA. <END>

NOTE 8.
DS1620 Digital Thermometer

The DS1620 is a digital thermometer. The chip internally has a temperature conversion table so the user does not have to make a separate conversion table. Temperature range between -55 and 125 degrees Celcius can be obtained by the DS1620 in units of 0.5 Degrees.



<Filename: ds1620.cul>

```
Const Device = CB280
Const iorst = 7
Const ioclk = 6
Const iodq = 5
Dim I As Integer

Delay 100
High iorst          ' init ds1620
Shiftout ioclk,iodq,0,12,8
Shiftout ioclk,iodq,0,3,8
Low iorst

Do
```

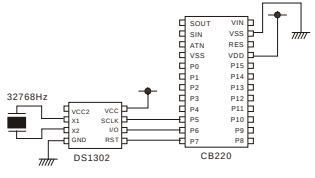


```
        High iorst
        Shiftout ioclk,iodq,0,&haa,8
        i = Shiftin(ioclk,iodq,4,9)
        i = i
        debug dec i,cr
        Low iorst
        Delay 100
Loop
```

The final value received can be divided into 2 to obtain the current temperature.
<END>

NOTE 9. DS1302 RTC

DS1302 RTC (Real Time Clock) is a chip that will acts as an electronic time keeper. It has the ability to keep time and date in real-time. We will show you how to implement this clock chip into your application in this note.



Pin	Function	I/O Direction	Explanation
RST	Reset	Input	Data transfer when High
SCLK	System Clock	Input	Clock signal
I/O	Data Input/Output	Input / Output	Data input/output

<Filename: ds1302.cul>

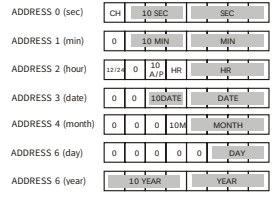
```
Const Device = CB220
      Const iorst = 7
      Const iodio = 6
      Const ioclk = 5
      Dim I As Integer
      Dim adr As Byte
      High iorst
      Shiftout ioclk,iodio,0,&h8e,8
      Shiftout ioclk,iodio,0,0,8
      Low iorst
      Delay 1
      High iorst
      Shiftout ioclk,iodio,0,&h80,8
      Shiftout ioclk,iodio,0,&H50,8
      Low iorst

      Do
            High iorst
            adr = &h81
            Shiftout ioclk,iodio,0,adr,8
            i = Shiftin(ioclk,iodio,4,8)
            Debug Hex i,cr
            Low iorst
            Delay 1000
      Loop
```


The above code will read ADDRESS 0, seconds value, and display it onto the DEBUG window.

At the beginning of the program, we will enable writes to the DS1302 chip and set the ADDRESS 0 to 50 seconds.

Within the Do Loop, we will read the data from DS1302. The DS1302 chip has 6 addresses as shown below:



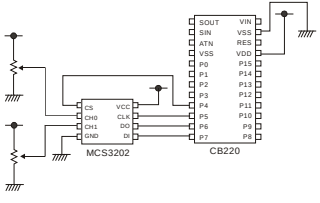
These addresses can be used to read and write to the DS1302.

Please note that the data is in BCD code format.

<END>

NOTE 10.
MCP3202 12 Bit A/D Conversion

The CUBLOC has a 10 bit A/D converter. Without a separate chip, you can get up to 10 bits of A/D conversion. But for greater resolution, meaning greater precision, you can use a chip like the MCP3202. MCP3202 is a 12 bit A/D converter that supports SPI protocol.
Here we will show you how to implement this 12 bit A/D converter into your project.



Pin	Function	I/O Direction	Explanation
CS	Chip Select	Input	Low for data communication
CLK	Clock	Input	Clock signal
DI	Data Input	Input	Data input from MCP3202
DO	Data Output	Output	Data output from MCP3202

<Filename: mcp3202.cul>

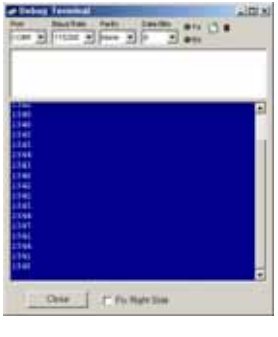
```
Const Device = CB280
Const iodi = 7
Const iodo = 6
Const ioclk = 5
Const iocs = 4
Dim I As Byte
Dim ad As Integer
Do
      Low iocs
```



```
      i = &b1011 'Channel 0
      'i = &h1111 'Channel 1
      Shiftout ioclk,iodi,0,i,4
      ad = Shiftin(ioclk,iodo,3,12)
      High iocs
      Debug Dec ad,cr
      Delay 100
Loop
```

The MCP3202 will convert voltage coming into CH0 and CH1 ports to a data value and retain it. The user can simply use SPI communication to read the value that the MCP3202 has converted.

The voltage inputted to the MCP320 CH0 and CH1 pins must not be greater than the voltage supplied to the MCP3202. The result of A/D conversion is displayed to the DEBUG window.



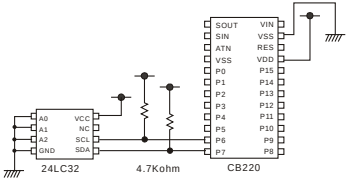
<END>

NOTE 11.
Read and write to the EEPROM

With the EEPROM, you can store between 0.5 to 64 KB of data. Data is retained even after powering off, allowing it to act almost as a small hard drive. If you want to retain a temperature setting for a temperature controller, you can simply store the value of the temperature in the EEPROM in case of power-outs.

CUBLOC has an internal EEPROM of 4KB. For small and simple data, you may use this internal EEPROM. In the case of larger data, you can use an EEPROM like 24LC512 to store up to 64KB of data.

Here we will show you how to access the 24LC32 4KB EEPROM through I2C protocol. The serial EEPROMs usually support either SPI or I2C. I2C EEPROMs name starts with 24XXXX and SPI EEPROMs name starts with 93XXX.



<Filename: eeprom.cul>

```
Const Device = CB280
Dim adr As Integer
Dim data As Byte
Dim a As Byte
data = &ha6
adr = &h3
Set I2c 7,6
Do
```



```
            I2cstart
            If I2cwrite(&b10100000)= 1 Then Goto err_proc
            a=I2cwrite(adr.byte1)
            a=I2cwrite(adr.lowbyte)
            a=I2cwrite(data)
            I2cstop
            Delay 1000
            I2cstart
            a=I2cwrite(&b10100000)
            a=I2cwrite(adr.byte1)
            a=I2cwrite(adr.lowbyte)
            I2cstart
            a=I2cwrite(&b10100001)
            a=I2cread(0)
            I2cstop
            Debug Hex a,cr
            ADR = ADR + 1
            DATA = DATA + 1
      Loop

err_proc:
      Debug "Error !"
      Do
      Loop
```

This example program will write a number to EEPROM and read from it. When this program runs correctly, numbers will increment on the DEBUG screen. You can easily modify this code to support other EEPROMs.

Note: Please wait at least 5ms after a write to the EEPROM.
<END>

NOTE 12.
XPORT Server program to control multiple devices from single PC

XPORT Custom Firmware Upload

The first thing to do is to upload the custom firmware to your XPORT.

1. Run DeviceInstaller
2. Choose the device you wish to recover/upgrade Firmware then click “Upgrade” (Here I selected an XPORT with address of 192.168.0.6)

3. Next Please choose “Create a custom installation…”and click “Next”

4. Please click “Browse”~

5. Please choose file called CUMAX_XXX.rom”. (This file is in the same directory as this PDF.)

6. Please click “Next”!

7. Please click “Next”again!

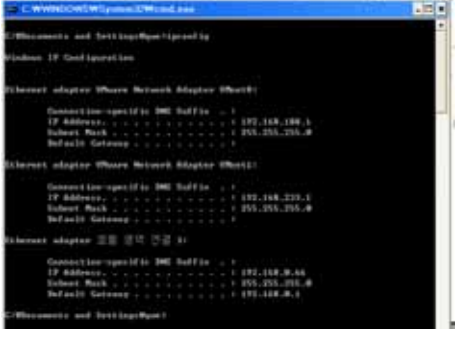
8. The custom XPORT firmware that works with CuMAX Server has been uploaded. Please wait a while and you will be able to see the XPORT after it resets itself.

You can use the above method to upload new versions of custom XPORT firmware or the original firmware too. (xpt03_XXX.rom)

First, we must find the IP address of the computer as this will serve as the IP address that XPORT will look for and send messages.

Please go to **Start->Run** and type “command”.

When the DOS prompt screen comes up, please type “ipconfig” and you will see the following screen.



If you will pay attention to where it says “Local Area Network” you will find your PC’s IP Address. Yes, I know mine is in different language. But as you can see, my PC’s IP address is **192.168.0.66.** Now we are ready to rock baby!

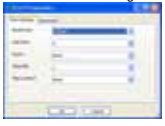
Okay, go back to the DeviceInstaller and find that XPORT you uploaded custom firmware to.

Now, select that XPORT and Click on “Configure”.

You see the following screen right?

Please click on “Ports”…

Click on Edit Settings~

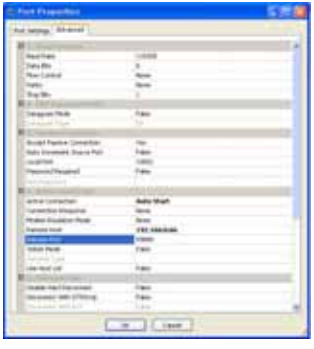


Please click on “Advanced” tab and you will see following screen. **There are TWO “Advanced” tabs. If you followed the directions clearly, you will be at the right “Advanced” tab like below.**

Here Please change “4. Active Connection” to “Auto Start”.

Please change Remote Host to your PC’s IP Address that we found earlier.
(Mine is 192.168.0.66.)

And change Remote Port to 59000 since CuMAX Server accepts connections on UDP port 59000.



If you have setup correctly, your screen should match the one above. Please Click " OK" and you Click " OK" again.

Later, when you are testing on the internet, you can change this address to the static IP address of the PC that the CuMAX Server will run on.

Please repeat above process for every XPORT that you want to control with the CuMAX Server program.

NOTE: You need to install Java software before using this program. Please go to www.java.com to download the free software.

Please run CUMAXvXXX.exe now.

You will see the above screen when you first start up the program. CuMAX Server will automatically search for the XPORTs.

How to Download an Object File

1. Compile and save to object file in CUBLOC Studio.

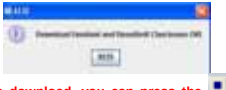
2. Simply select a file using the File-Open.

3. Select an object file.

2. Select the CUBLOC/CuTOUCH you want to download to.

4. Press the  run key.

5. If you get a message like below, you have successfully downloaded to your CUBLOC or CuTOUCH.



Note: Anytime during the download, you can press the  stop key to halt download.

The  stop key can also be used to reset your CUBLOC module when not downloading or monitoring.

How to Download a the Firmware

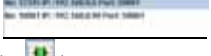
1. Select the CUBLOC/CuTOUCH you want to load a new firmware.

2. Click on the  button!
3. You will be able to see the status of the firmware download like below:

4. You will see a message like below if firmware was downloaded successfully.

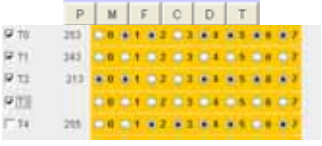
How to Monitor

1. Select the CUBLOC/CuTOUCH you want to monitor



2. Click on the monitor  button.
3. Use the checkmarks to monitor specific relays.

4. Use the relay buttons to monitor other relays.



5. Make sure to check on the checkmark below if you are using CuTOUCH or CB290.

Please refer to Max's Application on our forum: http://cubloc.com/phpBB2 for latest updates, downloads, and details.

ASCII CODE

Code	char.	Code	char.	Code	char.	Code	char.
00H	NUL	20H	SPACE	40H	@	60H	`
01H	SOH	21H	!	41H	A	61H	a
02H	STX	22H	"	42H	B	62H	b
03H	ETX	23H	#	43H	C	63H	c
04H	EOT	24H	$	44H	D	64H	d
05H	ENQ	25H	%	45H	E	65H	e
06H	ACK	26H	&	46H	F	66H	f
07H	BEL	27H	'	47H	G	67H	g
08H	BS	28H	(	48H	H	68H	h
09H	HT	29H	)	49H	I	69H	I
0AH	LF	2AH	*	4AH	J	6AH	j
0BH	VT	2BH	+	4BH	K	6BH	k
0CH	FF	2CH	,	4CH	L	6CH	l
0DH	CR	2DH	-	4DH	M	6DH	m
0EH	SO	2EH	.	4EH	N	6EH	n
0FH	SI	2FH	/	4FH	O	6FH	o
10H	DLE	30H	0	50H	P	70H	p
11H	DC1	31H	1	51H	Q	71H	q
12H	DC2	32H	2	52H	R	72H	r
13H	DC3	33H	3	53H	S	73H	s
14H	DC4	34H	4	54H	T	74H	t
15H	NAK	35H	5	55H	U	75H	u
16H	SYN	36H	6	56H	V	76H	v
17H	ETB	37H	7	57H	W	77H	w
18H	CAN	38H	8	58H	X	78H	x
19H	EM	39H	9	59H	Y	79H	y
1AH	SUB	3AH	:	5AH	Z	7AH	z
1BH	ESC	3BH	;	5BH	[	7BH	{
1CH	FS	3CH	<	5CH	\	7CH	|
1DH	GS	3DH	=	5DH	]	7DH	}
1EH	RS	3EH	>	5EH	^	7EH	~
1FH	US	3FH	?	5FH	_	7FH	DEL

Note for BASIC STAMP users

When using Parallax's Basic Stamp compatible development board, please be aware of the following:

There is a capacitor on the Basic Stamp compatible development boards which causes download error in CUBLOC Studio. Please short (or take out) the extra capacitor connected to the DTR of the board as shown below. CB220 has a this capacitor on the chip itself.