NetMedia

Basic Express BX-35
Application Note

Using CPUsleep for Power Down Mode

Power down mode

The BX-35 is capable of entering several sleep modes, one of which is a power down mode. During this mode, current is reduced to approximately 1.5 mA, depending on the system configuration. The processor can stay in this mode indefinitely, or it can be caused to wake up due to a watchdog timeout or a level interrupt. Here we'll discuss level interrupts.

Once the processor is in power down mode, a logic low can be applied to pin 17 in order to wake up the processor. The low level must be held for at least 16 ms. Alternatively, a hard reset can wake up and reboot the processor.

Software interface

The first step is to configure the INT1 pin (pin 17) to input-pullup:

```
Call PutPin(17, bxInputPullup)
```

In addition, the INT0 pin (pin 16) must also be configured as input-pullup. Code:

```
Register.DDRD  = Register.DDRD And bx1111_1011
Register.PORTD = Register.PORTD Or bx0000_0100
```

The next step is to create a task that executes a WaitForInterrupt procedure. WaitForInterrupt blocks the task until an interrupt occurs. In this case, the interrupt is triggered by a logic low on pin 17. This line should appear in the task:

```
Call WaitForInterrupt(bxPinLow) ' This should be in a separate task.
```

Now we need to configure sleep-related bits in the MCUCR register:

```
' Clear SM0 (bit 4).
Register.MCUCR = Register.MCUCR And bx1110_1111

' Set SE (Sleep Enable, bit 6). Also set SM1 (bit 5) for
' power down mode.
Register.MCUCR = Register.MCUCR Or  bx0110_0000
```

The last step is to call CPUsleep, which executes a special internal sleep function:

```
Call CPUsleep
```

At this point, the processor goes into power down mode. Current drops to about 1.5 mA, depending on the system configuration. A level interrupt will wake up the processor and restart execution on the line following CPUsleep.

Warning -- you should insure that a WaitForInterrupt is pending before calling CPUsleep. Otherwise a level interrupt may not wake up the processor. Note that if pin 17 is *already* low before calling WaitForInterrupt, the interrupt will occur immediately. If this happens, you should refrain from calling CPUsleep.

Also, if you're using a serial port to transmit data, make sure any output queues are emptied before calling CPUsleep. Otherwise the program may hang. In addition, data may be garbled, since the processor stops transmission immediately upon entering a sleep mode. Similar precautions should be taken with input queues.

Code example

A separate example program is provided. The program is in file PowerDownExample.bas. Low level details can be found in module LowPower, which includes a separate task for calling WaitForInterrupt.