

USB-CTR04 USB-CTR08 Counter Output using Gate

The following example demonstrates how to output two signals using the counter output pins. The counter gate inputs are used start / stop the output.

This is a CSharp program that uses the MCCDAQ dot net API. The complete Visual Studio 2008 project is attached to this article.

To access the API, add a reference to the MccDaq object. Adding the reference is usually accomplished by right clicking the Project [under the Project Explorer] and selecting Add Reference. The complete project is attached to the article and was created with Visual Studio 2008 using CSharp. Compiling and execution is quick as there is minimal Windows overhead and associated code.

To make the code more readable a few convenience functions were added such as IsError, GetBoardNum, and WaitForKey. The IsError function checks the error number in the ErrorInfo object and if not zero displays the error message. The GetBoardNum function begins reading the Device strings from each number location and as soon as it finds one that contains the identifying string it exits. The WaitForKey does just that - waits for someone to press a key. To view the complete program project download the attachment below.

Disclaimer:

The attached Code or Example is provided As Is. It has not been tested or validated as a product, for use in a deployed application or system, or for use in hazardous environments. You assume all risks for use of the Code or Example.

```
using System;
using MccDaq; // also, add reference to MccDaq to the project

namespace PeriodMode
{
    class Program
    {
        public const string DEVICE = "USB-CTR"; //works for USB-CTR04 or USB-CTR08

        public const int OutTimerNum = 0; //Frequency source - connect to counter inputs 0
& 1

        public const CounterMode Mode = CounterMode.RangeLimitOn|
            CounterMode.OutputInitialStateLow|
            CounterMode.OutputOn|
            CounterMode.GatingOn;

        /*////////////////////////////////////*/
```

USB-CTR04 USB-CTR08 Counter Output using Gate

```
static void Main(string[] args)
{
    int BoardNum    = 0;
    double freq     = 1000000.0f; //1MHz
    double duty     = 0.50f;
    double intDelay = 0;
    int CounterNum  = 0;
    int MapCounter  = 0;

    Console.WriteLine("Locating Device...Please wait\n");

    BoardNum = GetBoardNum(DEVICE);

    if (BoardNum == -1)
    {
        Console.WriteLine("No {0} detected!", DEVICE);
        WaitForKey("Press any key to exit\n");
        return;
    }
    else
    {
        MccBoard daq = new MccDaq.MccBoard(BoardNum);

        //Digital bit D0 controls the counter gates which enable the counters

        IsError(daq.DConfigPort(DigitalPortType.AuxPort,DigitalPortDirection.DigitalOut));
        //set D0 low so to disable counting
        IsError(daq.DOut(DigitalPortType.AuxPort, 0));

        //instruct the counter to count to 1000 and each time it reaches 100
        //output a level high until count is 200
        CounterNum = 0;
        IsError(daq.CConfigScan(CounterNum,
                                Mode,
                                CounterDebounceTime.Debounce500ns,
                                CounterDebounceMode.TriggerAfterStable,
                                CounterEdgeDetection.RisingEdge,
                                CounterTickSize.Tick20pt83ns,
                                MapCounter));

        IsError(daq.CLoad32(CounterRegister.MinLimitReg0, 0)); //count from 0 to 999
        or 1000uS
    }
}
```

USB-CTR04 USB-CTR08 Counter Output using Gate

```
IsError(daq.CLoad32(CounterRegister.MaxLimitReg0, 999));

IsError(daq.CLoad32(CounterRegister.OutputVal0Reg0, 1)); //set to one so that
initial state is low
IsError(daq.CLoad32(CounterRegister.OutputVal1Reg0, 100));//100uS after start
set counter output low

//instruct the counter to count to 1000 and each time it reaches 800
//output a level high until count is 900
CounterNum = 1;
IsError(daq.CConfigScan(CounterNum,
                        Mode,
                        CounterDebounceTime.Debounce500ns,
                        CounterDebounceMode.TriggerAfterStable,
                        CounterEdgeDetection.RisingEdge,
                        CounterTickSize.Tick20pt83ns,
                        MapCounter));

IsError(daq.CLoad32(CounterRegister.MinLimitReg1, 0)); //count from 0 to 999
or 1000uS
IsError(daq.CLoad32(CounterRegister.MaxLimitReg1, 999));

IsError(daq.CLoad32(CounterRegister.OutputVal0Reg1, 401));//at 401uS set
counter output high
IsError(daq.CLoad32(CounterRegister.OutputVal1Reg1, 500));//100uS later set
it low

//connect timer output 0 to counter input 0 & 1 so that counter have something
to count
IsError(daq.PulseOutStart(OutTimerNum,
                          ref freq,
                          ref duty, 0,
                          ref intDelay,
                          IdleState.Low,
                          PulseOutOptions.Default));

WaitForKey("Press any key to start output\n");
//set D0 high to enable counting
IsError(daq.DOut(DigitalPortType.AuxPort, 1));

WaitForKey("Press any key to exit\n");
```

USB-CTR04 USB-CTR08 Counter Output using Gate

```
        IsError(daq.DOut(DigitalPortType.AuxPort, 0)); //set D0 back low
        IsError(daq.PulseOutStop(OutTimerNum)); //disable frequency source
        IsError(daq.CClear(0)); //clear counter 0
        IsError(daq.CClear(1)); //clear counter 1
    }
}
/*////////////////////////////////////////////////////////////////////////////////////////////////////////////////*/
public static int GetBoardNum(string dev)
{
    for (int BoardNum = 0; BoardNum < 99; BoardNum++)
    {
        MccDaq.MccBoard daq = new MccDaq.MccBoard(BoardNum);
        if (daq.BoardName.Contains(dev))
        {
            Console.WriteLine("board number = {0}\n", BoardNum.ToString());
            return BoardNum;
        }
    }
    return -1;
}
/*////////////////////////////////////////////////////////////////////////////////////////////////////////////////*/
public static void WaitForKey(string s)
{
    Console.WriteLine(s);
    do
    { //idle loop
        System.Threading.Thread.Sleep(10);
    } while (!Console.KeyAvailable);
    Console.ReadKey();
}
/*////////////////////////////////////////////////////////////////////////////////////////////////////////////////*/
public static int IsError(ErrorInfo e)
{
    if (e.Value != 0)
    {
        Console.WriteLine(e.Message);
    }
}
```

USB-CTR04 USB-CTR08 Counter Output using Gate

```
    WaitForKey("Press any key to continue\n");  
    return 1;  
  }  
  return 0;  
}  
  
/*////////////////////////////////////*/  
  
}  
}
```

Measurement Computing Data Acquisition Knowledgebase
<https://kb.mccdaq.com/KnowledgebaseArticle50577.aspx>