KMTronic C# API driver



In Extended Data you can find the driver for the KMTronics



It is a very simple API code

<https://www.kmtronic.com/usb-relays.html/USB-Relay-Controller-Eight-Channel.html>

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.IO.Ports;  using System.Linq;  using System.Management;  using System.Text;  using System.Threading.Tasks;  namespace KMTronicsApi  {    public class KMTronics  {  SerialPort m\_serialPort = new SerialPort();  public KMTronics()  {  }  public KMTronics GetBase()  {  return this;  }  string GetUSB\_ISS\_PortName()  {  using (var searcher = new ManagementObjectSearcher  ("SELECT \* FROM WIN32\_SerialPort"))  {  string[] portnames = SerialPort.GetPortNames();  var ports = searcher.Get().Cast<ManagementBaseObject>().ToList();  var tList = (from n in portnames  join p in ports on n equals p["DeviceID"].ToString()  select n + " - " + p["Caption"]).ToList();  int i = 0;  foreach (string s in tList)  {  if (s.Contains("KMTronic USB 8 Relays") == true)  {  string[] words = s.Split(new Char[] { '-' });  return words[0].Trim();  }  i++;  }  }  return string.Empty;  }  public virtual void Write(byte relay, byte onoff)  {  if (relay < 1 || relay > 8)  {  throw (new SystemException("Relay is between 1 -8"));  }  try  {  m\_serialPort.Write(new byte[] { 0xFF, relay, onoff }, 0, 3);  }  catch (Exception err)  {  throw (new SystemException("Kmtronic : " + err.Message));  }  }  public virtual void Write(byte relay, bool onoff)  {  if (relay < 1 || relay > 8)  {  throw (new SystemException("Relay is between 1 -8"));  }  try  {  byte val = (byte)(onoff == true ? 1 : 0);  m\_serialPort.Write(new byte[] { 0xFF, relay, val }, 0, 3);  }  catch (Exception err)  {  throw (new SystemException("Kmtronic : " + err.Message));  }  }  public virtual void On(byte relay)  {  try  {  m\_serialPort.Write(new byte[] { 0xFF, relay, 1 }, 0, 3);  }  catch (Exception err)  {  throw (new SystemException("Kmtronic : " + err.Message));  }  }  public virtual void Off(byte relay)  {  try  {  m\_serialPort.Write(new byte[] { 0xFF, relay, 0 }, 0, 3);  }  catch (Exception err)  {  throw (new SystemException("Kmtronic : " + err.Message));  }  }  public virtual bool Open(string comPort, bool auto)  {  try  {  if (auto == true)  {  comPort = GetUSB\_ISS\_PortName();  }  m\_serialPort.Close();  m\_serialPort.PortName = comPort;  m\_serialPort.WriteTimeout = 2000;  m\_serialPort.ReadTimeout = 2000;  m\_serialPort.Open();  if (m\_serialPort.IsOpen)  {  return true;  }  else  {  return false;  }  }  catch (Exception err)  {  throw (new SystemException("Kmtronic : " + err.Message));  }  }  public virtual void Close()  {  try  {  m\_serialPort.Close();  }  catch (Exception err)  {  throw (new SystemException("Kmtronic : " + err.Message));  }  }  }  } |