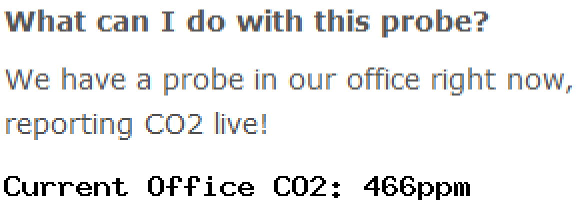
AN112 – Simple Web Application of the K30 Probe

The K30 Probe can be used in many different applications as a quick way to start collecting environmental CO2 data. We will show you a simple application of the probe to report data, via a HTTP GET request, to a remote PHP-based application, and ultimately display that data on a website, as we have done on the probe's product page:



What can I do with this probe?

We have a probe in our office right now,
reporting CO2 live!

Current Office CO2: 466ppm

Server Setup

You will need a web-host that supports PHP scripting and FTP access to your hosting account. Almost all modern hosts meet these criteria. We will create two files on the server, one called image.php, that will generate the image of the current CO2, and one called update.php, which will process updates from the probe sent from a remote computer.

Image.php

```
<?php
Header ("Content-type: image/gif");
$textfile = "co2.txt";

$string = file($textfile);


$font = 5;
$width = ImageFontWidth($font)* strlen("Current Office CO2: " . $string[0] . "ppm");
$height = ImageFontHeight($font);
$im = ImageCreate($width,$height);

$x=imagesx($im)-$width ;
$y=imagesy($im)-$height;
$background_color = imagecolorallocate ($im, 242, 242, 242); //white background
$text_color = imagecolorallocate ($im, 0, 0,0);//black text
$trans_color = $background_color;//transparent colour
imagecolortransparent($im, $trans_color);
imagestring ($im, $font, $x, $y, "Current Office CO2: " . $string[0] . "ppm",
$text_color);

imagegif($im);
ImageDestroy($im);
```

Update.php

```
<?php
header(" ");
if($_GET['key'] == "co2passwordhere")
{
    $file = ("co2.txt");
    $fp = fopen($file , "w");
    fputs($fp , $_GET['co2']);
    fclose($fp);
}
```

The update file simply takes two parameters from the HTTP request, key and co2. Key represents a measure of security; a unique passphrase is chosen to ensure that only authorized sources can update the CO2 value. CO2 is simply a ASCII string representation of the current CO2 in the room. It is written to a file, co2.txt, where it will be read by image.php. Image.php uses the built-in PHP image library to generate an image, streamed to the client's browser, after reading the CO2 value from the written file.

Client Setup

The client in this situation will be a simple C# application. Create a Console application and within the static main body, paste the code on the following page. Change the COM port in the code to reflect the COM port used by the probe when the FTDI drivers were installed upon connection.

Bringing it all Together

Once the C# application is running you should be able to see CO2 data being displayed in the console window. At the same time these readings are being reported to the web service. Requesting image.php should yield a graphic similar to the one displayed on CO2Meter.com.

The applications of the K30 probe can be extended far beyond simply displaying the CO2. At this point it would be trivial to add a database for permanent storage, or support for multiple probes, to the application. We hope this application note provides a starting point for one to develop custom applications using the probe.

```
SerialPort sp1 = new SerialPort("COM3", 9600);
sp1.Open();

while (true)
{
    // UART COM string from MODBUS Appendix.
    sp1.Write(new byte[] { 0xFE, 0x04, 0x00, 0x03, 0x00, 0x01, 0xD5, 0xC5 }, 0, 8);

    // Wait to read data back.
    while (sp1.BytesToRead < 7)
    {
        System.Threading.Thread.Sleep(100);
    }

    // Read the CO2 Values
    byte[] buffer1 = new byte[7];
    sp1.Read(buffer1, 0, 7);

    // Swap byte order
    byte[] buffer2 = new byte[2];
    buffer2[0] = buffer1[4];
    buffer2[1] = buffer1[3];

    double co2K30 = BitConverter.ToInt16(buffer2, 0);

    try
    {
        // Make the GET request.
        System.Net.HttpWebRequest PageRequest =
(System.Net.HttpWebRequest)(System.Net.WebRequest.Create("http://somehost.com
/update.php?key=co2passwrd&co2=" + co2K30));
        PageRequest.GetResponse();
    }
    catch
    {

    }
    Console.WriteLine("read: {0}", co2K30);

   System.Threading.Thread.Sleep(5000);
}
```