BERGONLINE.DK
coding made easier!

howto: Netduino - read temperature from DS18B20

14. July 2013 13:04 by Steffen Berg in C#, Netduino // Tags: Netduino, c-Sharp, c#, Development, DS18B20, temperature // Comments (0)

How to connect the DS18B20?

It is actually pretty easy to connect the DS18B20 temperature sensor to your Netduino device. In this example I use an waterproof sensor ordered at adafruit.com (link: http://www.adafruit.com/products/381).

The sensor has 3 wires:

- red (+5v)

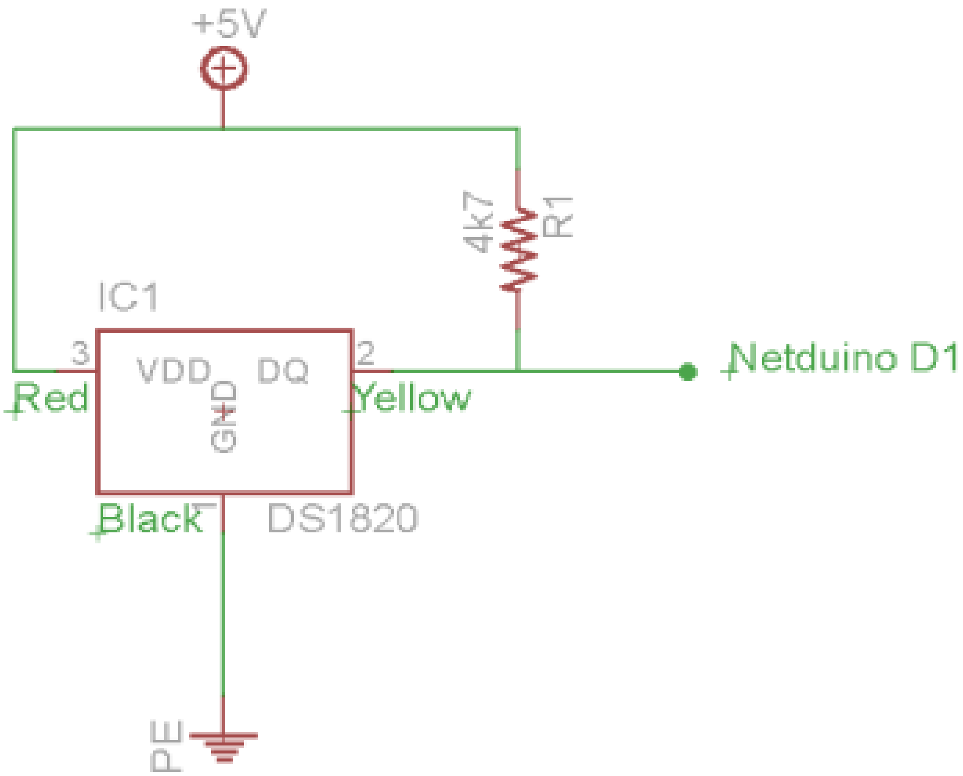
- yellow (Digital output)

- Black (ground)

The black wire will be connect directly to GND, the red connected directly to +5v and the yellow to D1 (digital I/O port 1 at Netduino). The last thing we need to connect is a pull-up resistor for the yellow wire, to this we use a 4k7 ohm resistor connected to both the yellow and the red wire.

How to read temperature with Netduino?

The code below is made for reading temperature from multiple DS18B20 sensors on one onewire port. The code starts by finding all the devices and then iterate them. By setting the TemperatureFormat on the DS18B20 class, you are able to switch between celcius and fahrenheit.



```
ystem;
ystem.Threading;
icrosoft.SPOT;
icrosoft.SPOT.Hardware;
ecretLabs.NETMF.Hardware.Netduino;

ce Temperature

lic class Program

 public static void Main()
 {
     MultipleDevices();
 }

 private static void MultipleDevices()
 {
     OneWire oneWire = new OneWire(new OutputPort(Pins.GPIO_PIN_D0, false));

     var devices = oneWire.FindAllDevices();

     while (true)
     {
         if (devices.Count > 0)
         {
             foreach (var device in devices)
             {
                 oneWire.TouchReset();
                 oneWire.WriteByte(DS18B20.MatchROM);        // Match ROM, we have multiple devi
                 DS18B20.SetDevice(oneWire, device);         // Set device
                 oneWire.WriteByte(DS18B20.ConvertT);        // Start temperature conversion
                 while (oneWire.ReadByte() == 0) ;           // wait while busy

                 oneWire.TouchReset();
                 oneWire.WriteByte(DS18B20.MatchROM);        // Match ROM
                 DS18B20.SetDevice(oneWire, device);         // Set device
                 oneWire.WriteByte(DS18B20.ReadScratchpad);  // Read Scratchpad

                 double temp;
                 temp = DS18B20.GetTemperature((byte)oneWire.ReadByte(), (byte)oneWire.ReadByte(
                 Debug.Print("Device: " + devices.IndexOf(device) + ", temperature: " + temp + "
             }
         }
         else
         {
             Debug.Print("No device detected");
         }

         Thread.Sleep(1000);
     }
 }
```

```
lic static class DS18B20

 public const byte FamilyCode = 0x28;

 // Commands
 public const byte ConvertT = 0x44;
 public const byte CopyScratchpad = 0x48;
 public const byte WriteScratchpad = 0x4E;
 public const byte ReadPowerSupply = 0xB4;
 public const byte RecallE2 = 0xB8;
 public const byte ReadScratchpad = 0xBE;
 public const byte MatchROM = 0x55;
 public const byte SkipROM = 0xCC;

 // Fields
 public static TemperatureFormat TemperatureFormat;

 // Get temperature
 public static float GetTemperature(byte tempLo, byte tempHi)
 {
     float ret = ((short)((tempHi << 8) | tempLo)) / 16F;
     if (TemperatureFormat == TemperatureFormat.Fahrenheit)
     {
         ret = (ret * 9 / 5) + 32;
     }
     return ret;
 }

 // Set device
 public static byte[] SetDevice(OneWire oneWire, object device)
 {
     byte[] b = (byte[])device;
     foreach (var bTmp in b)
     {
         oneWire.TouchByte(bTmp);
     }
     return b;
 }


lic enum TemperatureFormat

 Celcius,
 Fahrenheit
```

You are now ready for reading temperatures. Combining it with a MicroSD Card and you have your own temperature logger :)

Top