Make a Thermometer and Hygrometer using Arduino and C#

Vulsan Bianca Maria

Student at Technical University of Cluj-Napoca, Computer Science, Group 30414

Email: Bianca.Vulsan@student.utcluj.ro

**Abstract.** The aim of this project is to collect data via a temperature and humidity sensor and display it in an Android application using C#. The main reason that such a project is useful is that it provides a handy method of measuring the temperature in the room and the humidity of the air. Furthermore, it is a wonderful way to acquire some important knowledge of working with sensors using Arduino and transmitting data via bluetooth to an C# app. The most segnificant conclusion I have drown at the end of this project is that it is more satysfing to use somethig that you have done and understand how it works than just taking it for granted, because just by understanding other’s ideas someone can evolve and come up with its own.

1. **Introduction**

First of all, any project starts with establishing the list of components needed. So, for this project were used the folowing elements:

* DHT11 temperature and humidity sensor
* Arduino Uno board
* A breadboard
* HC-05 bluetooth module
* Wires to connect the components
* An 1 Ohm resistor and a 2 Ohms resistor for the bluetooth module (optional)
* Visual Studio or other application for writing C# code
* Arduiono application installed

In the following section there will be presentet step by step the way everything come together.

1. **DHT11 Temperature and Humidity Sensor**

The DHT11 is a digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin. An important aspect is that it requires careful timing to collect data, as one can get new data from it once every 2 seconds. The specifications for this sensor are listed below:

* Supply Voltage: +5 V
* Temperature range :0-50 °C error of ± 2 °C
* Humidity :20-90% RH ± 5% RH error
* Interface: Digital
  1. The physical concept

The protocol DHT11 uses is “unusual”. It transmits the bits very fast as short impulses. As Arduino does not have an operation system, it functionate closer to the real time than Raspberry Pi and reads more easily the impulses. But, even so, the information must be encoded in a special way, because the embedded function pulseIn() is not fast enough.

DHT11 transmits 5 bytes packets. So, taking into consideration that each byte has 8 bits, the packet has 5\*8 = 40 bits.

* 1. Connection for Arduino

|  |
| --- |
| * 1. A screenshot of a computer       Description automatically generated with medium confidence |
| **Figure 1.** DHT11 connection. |

1. **HC-05 Bluetooth Module**

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth*.* It has six pins: Key/EN, VCC, GND, TXD (Transmit Serial Data), RXD (Receive Serial Data) and State pin.

* 1. The physical concept

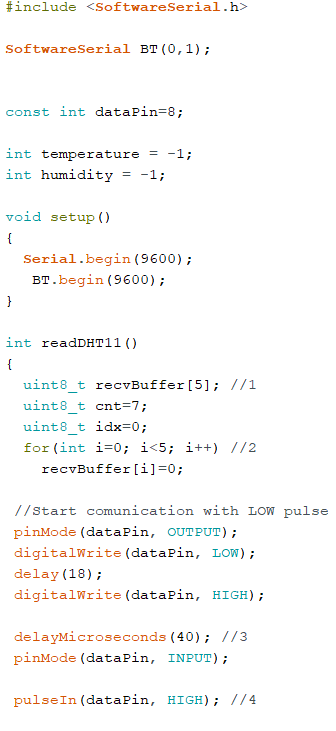
HC-05 Chip is known as Bluetooth SPP (Serial Port Protocol) module. It is Used for transparent wireless serially connecting setup. It is fully dedicated Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation Along with complete 2.4GHz radio transceiver. It has CSR Blue core 04-External single chip Bluetooth system with Complementary Metal Oxide Semiconductor technology and with AFH (Adaptive Frequency Hopping Feature). It has the Dimensions as small as 12.7mmx27mm.

* 1. Connection for Arduino

|  |  |
| --- | --- |
| A picture containing text, electronics, circuit  Description automatically generated | **Figure 1.** HC-05 connection. |

1. **Code for Arduino**

After understanding how the sensor and the bluetooth module work it is necessary to implement the code in order to let them communicate with the computer and display the data received.

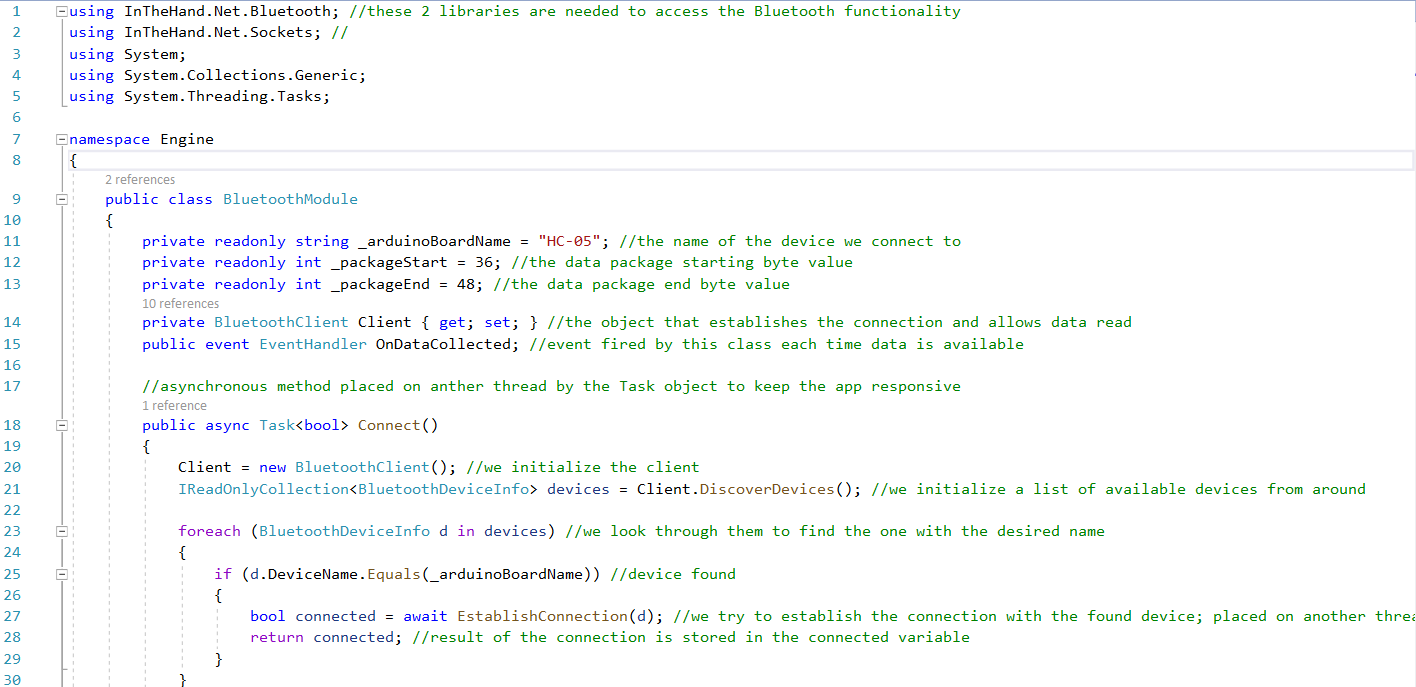
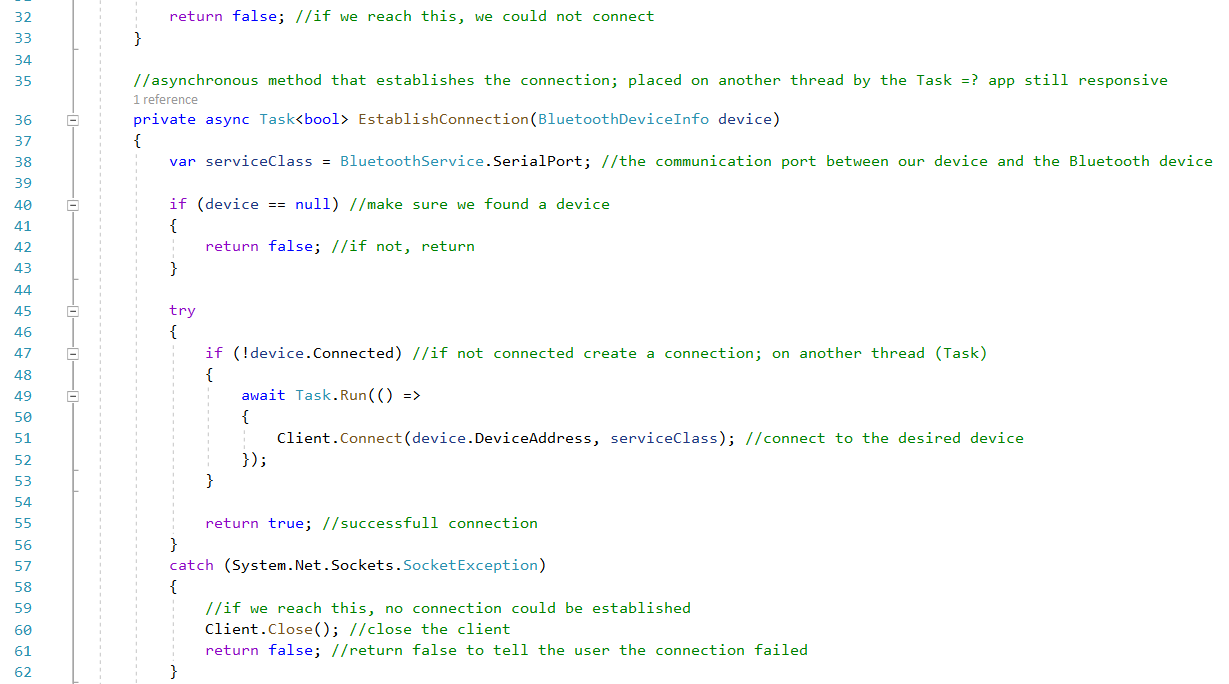
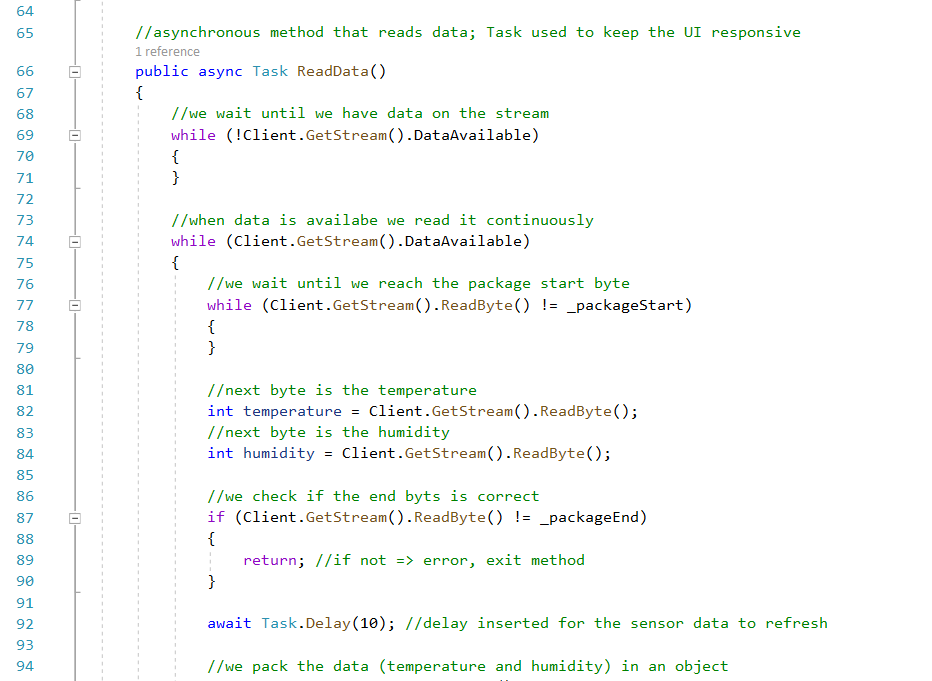
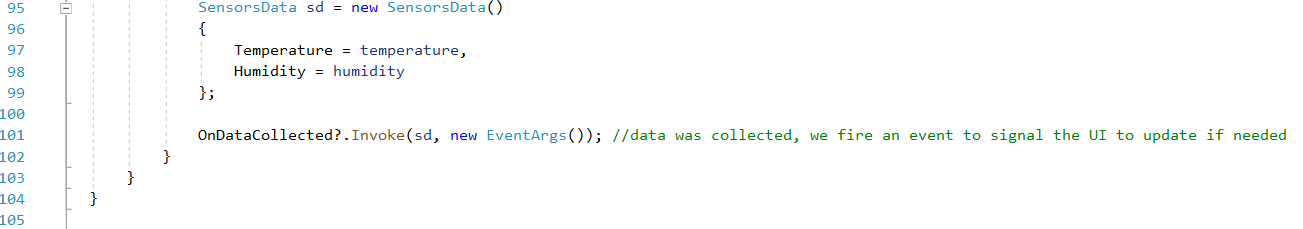






1. **The Application**
   1. C# code

Because it is more convenient to see the data on the phone, displayed in a more attractive way, we will send it to a C# Android application using the following code:

* 1. 
  2. 
  3. 
  4. 
  5. Interface of the application

|  |
| --- |
|  |
| **Figure 3.** Interface. |

1. **Conclusions**

The main thing that I have learned by doing this project is that doing something by yourself and understanding how it works is more fascinating and productive than buying it. Besides all the satisfaction getting when it works there are many other skills implied. For example, I’ve learned some very interesting aspects regarding the sensor used and the Bluetooth module, I’ve understood better specific features of C# programming language and finally I’ve increased my attention to details.

Figure 4 shows the physical part of the project. I hope that in the future I will integrate it in a box to improve the aspect.

|  |
| --- |
| A picture containing text  Description automatically generated |
| **Figure 4.** Physical part of the project. |

1. **References**

[1] Tero Karvien, Kimmo Karvien, Ville Vatokari 2017 Să construin senzori: proiecte si experimente pentru a masura lumea cu placile Arduino si Raspberry Pi (București: M.A.S.T.)

[2] electronicslovers.com/2016/08/how-to-interface-hc-05-bluetooth-module.html

[3] https://www.arduino.cc/en/Guide