ΕΞΩΦΥΛΟ

Acknowledgements

## Abstract

Contents

[Abstract 2](#_Toc394269172)

[1. Introduction 6](#_Toc394269173)

[1.1. Motivation 6](#_Toc394269174)

[1.2. Purposes and targets 6](#_Toc394269175)

[1.3. Project Structure 6](#_Toc394269176)

[2. Implementation Methodology 6](#_Toc394269177)

[3. Project Plan 7](#_Toc394269178)

[3.1. State Of The Art 7](#_Toc394269179)

[4. Basic Part 8](#_Toc394269180)

[4.1. Hardware Components 8](#_Toc394269181)

[4.1.1. Arduino 9](#_Toc394269182)

[4.1.2. Sensors 12](#_Toc394269183)

[References 12](#_Toc394269184)

Nomenclature

* MC / MCU Microcontroller
* RH Relative Humidity
* °C Degrees Celsius

[Figure 1 - The Arduino Mega 2560 R3 Microcontroller Board 9](file:///D:\Dropbox\GraduationalProject\SenseBox\Report.docx#_Toc394268236)

[Figure 2 - The Arduino Ethernet Shield 11](#_Toc394268237)

# Introduction

The purpose of this project is to measure the weather and virtualize the data. The data gathering process is implemented upon Arduino, an open source hardware prototyping platform, which allows an easy implementation of sensors and interactive elements. Data gathered by Arduino are stored on a database and on the next level these data are used to make graphs on different platforms. Also the project included a live weather camera that rotates to spot almost everywhere the user wants.

## Motivation

Building my own weather station was an idea that came into my mind about two years ago. I first thought of building it for educational purposes and as a hobby. Later on I realized that the project could also serve as my final year’s project so I declared it.  
  
My interested on weather data has been inherited from my favorite sport – Sailing. For sailors observing the weather is a major part of their daily routine so this was the icing on the cake to begin!

## Purposes and targets

Sdasdasd

## Project Structure

The project is split in four parts.

1. The Arduino part in which data are gathered.
2. The server part that hosts the server-side PHP code that is used to store and retrieve data from the database.
3. The Android Application part that visualizes the data.
4. The Web Site part that also visualizes the data.

# Implementation Methodology

Cvsdfsdf

# Project Plan

The project has been implemented using the latest technologies available for its needs. The next chapter state of the art provides a more in depth analysis about the background of the project.

## State Of The Art

* **Open – Source Hardware**

Open - Source has become a popular expression, but mostly with regard to software. The principles and definitions of Open-Source Hardware (OSHW) are closely related to those of Open-Source Software (OSS) from the Open Source Initiative [1, 2]. Interested people can study, modify, distribute, make and sell designs based on those of OSHW products [3]. Through open development, people get the possibility to learn and understand how OSHW works, so that they are able to control and modify their technology. Machines, devices, or other physical things produced under the OSHW license must comply with the definitions of the Open Source Hardware Association [1].

* **Arduino Platform**

Arduino is a prototyping platform containing a microcontroller board (MC) as core element, a programming language and an integrated development environment (IDE) [4]. OSHW MCs from Arduino are based on ATmega8, ATmega168, ATmega2560 and latest boards on [ATmega32u4](http://arduino.cc/en/Main/ArduinoBoardYun), [Sitara AM335x](http://arduino.cc/en/Main/ArduinoBoardTre). They can be used to develop interactive prototypes, which are using input from sensors to control output devices connected to the same board. The capabilities of the MCs allow several methods for in- and outputs of signals.

The simplified IDE with a wire-based programming language also allows beginners to realize complex projects in short time. Documentation of the IDE and the language reference can be found on the Arduino homepage [5]. All code samples are released into the public domain. Additionally, the language can be extended with C++ libraries to enable more functionality. All official software tools from Arduino are published under the OSS license and are platform-independent.

The well documented Arduino hard- and software makes it even possible to rebuild MCs by oneself. However, pre-assembled Arduino boards are relatively inexpensive compared to other commercial MC platforms available on the market

* **Raspberry PI**

The Raspberry PI is a credit – card – sized single board computer developed mainly for educational purposes. As core element it has a Broadcom BCM2835 system on a chip (SoC) [9] which includes an ARM11 family 700MHz processor, a VideoCore IV GPU and 512MB RAM.

The Raspberry PI foundation provides a lightweight version of Debian (Raspbian) and Arch Linux ARM distribution as official operating systems.

Tools are available for Python as the main programming language, C, Java and Perl.

For the needs of the project, the RPI is used as server, hosting the script files that are used to store and retrieve weather data in the database as well as hosting the web page. Later it is explained why the RPI is the perfect small server.

* **Automated Weather Stations**

Automated weather stations (AWSs) are meteorological stations, which perform climatic observations and data transmission automatically [6]. Main advantages compared to human weather observations are, that measurements, read out by a central data-acquisition unit, are more reliable and can be performed much more frequently. Sensors of an AWS are operated by a microprocessor. It allows exact sampling of sensor data and processing for averaging or filtering of the samples. The result will be a series of observations which are representative over a limited area.

* **Android Platform**

Android is a mobile operating system based on the Linux Kernel that is currently developed by Google. It powers hundreds of millions of mobile devices around the world. It is the largest installed base of any mobile operating system platform.

Android’s source code is released by Google under open source licenses (Apache License 2.0 and GNU General Public License 2.0 [7, 8]).

Its open nature has encouraged a large community of developers and enthusiasts to use the open – source code as a foundation for community-driven projects, which adds new features for advanced users.

* **Google Chart API**

The Google Chart API [10] is a tool that lets people easily create a chart from some data and embed it in a web page. Google creates a [PNG](http://en.wikipedia.org/wiki/Portable_Network_Graphics) image of a chart from data and formatting parameters in an [HTTP](http://en.wikipedia.org/wiki/HTTP) request. Many types of charts are supported, and by making the request into an image tag, people can simply include the chart in a web page.

# Basic Part

//TODO REQUIREMENT ANALYSIS

## Hardware Components

The Microcontroller controls the data storage process and sampling frequency. Sampling rates of sensors are depending on the time constant value that we set. Sensors and Microcontroller functionality is explained in depth in the upcoming section.

### Arduino

The Arduino and the Ethernet Shield can be imagined as the ‘brain’ of the weather station. It provides all the core functionality from sensor manipulation up to data logging. The Arduino Mega and the Arduino Ethernet Shield are introduced below.

#### Arduino Mega 2560 R3

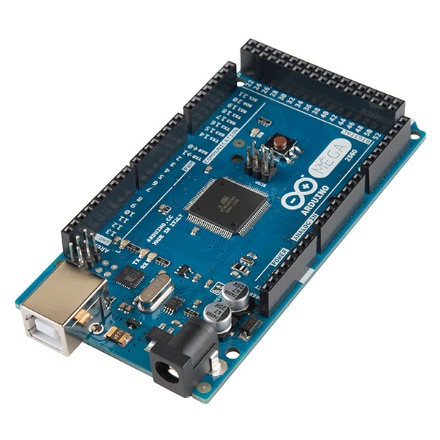


Figure 1 - The Arduino Mega 2560 R3 Microcontroller Board

Arduino Mega 2560 [11] is the board that is powered by ATmega2560. Its main advantages compared to other Arduino boards are the clock speed of 16MHz, the flash memory of 256Kbytes of the ATmega2560 chip and the 8Kbytes SRAM. The large flash and SRAM memories offer the Arduino vast of space for more interactive components to be added in future improvements.

The Arduino Mega provides 54 digital pins. They can be used as input or output and are operating at 5V. Each pin can provide or receive a maximum of 40mA and has an internal pull-up resistor of 20 – 50 kΩ. The pull-up resistor is disconnected by default but they can be accessed from software by setting the **pinMode()** as **INPUT\_PULLUP**. This effectively inverts the behavior of the INPUT mode, where HIGH means the sensor is off, and LOW means the sensor is on. In addition, some pins have specialized functions:

* **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).**

Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

* **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).**

These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

* **PWM: 2 to 13 and 44 to 46.**

Provide 8-bit PWM output with the [analogWrite()](http://arduino.cc/en/Reference/AnalogWrite) function.

* **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).**

These pins support SPI communication using the SPI library.

* **LED: 13.**

There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

* **TWI: 20 (SDA) and 21 (SCL).**

Support TWI communication using the [Wire library](http://arduino.cc/en/Reference/Wire).

(BMP085 sensor uses the Wire protocol to communicate with Arduino)

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 210 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

#### Arduino Ethernet Shield

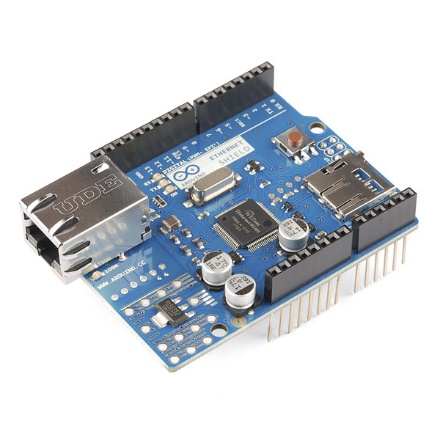


Figure 2 - The Arduino Ethernet Shield

The Ethernet shield [12] is powered by a Wiznet W5100 ethernet controller and also includes a micro SD card slot for storage purposes. It is made exclusively for Arduino boards and connects over the SPI port.

The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

It worth mentioning that pins SPI pins cannot be used as digital in or output any more when using the Ethernet Shield as they are being used by the SPI bus. Pins 4, 10 are used for the SD card and cannot be used as well. All the other pins are free to be used.

The Ethernet Shield is used in the project for data logging and as server for controlling the pan-tilt camera mount.

To integrate functionalities of the Ethernet Shield into the whole system, Ethernet [13] and SD [14] libraries are used.

### Sensors

In this section the functionality and the characteristics of each sensors used by the prototype will be analyzed. In addition, connection and communication examples are given in this section.

#### Relative Humidity and Air Temperature (DHT22 or RHT03)

The RHT03 sensor from is made of two parts, a capacitive humidity sensor and a thermistor. There is also a basic chip inside that does some analog to digital conversion and offers a digital signal with the temperature and humidity.

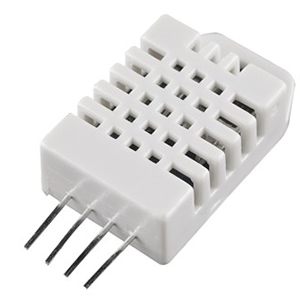


Figure 3 - RHT03

The RHT03 was chosen because of its high Resolution and Accuracy. Technical Specifications are shown in the table below [15].

|  |  |
| --- | --- |
| **Model** | **DHT 22 also known as RHT03** |
| Power Supply | 3.3 – 6V DC |
| Output Signal | Digital Signal via MaxDetect 1-wire bus |
| Sensing Element | Polymer Humidity Capacitor |
| Operating Range | Humidity: 0 – 100% RH  Temperature: - 40 ~ 80 °C |
| Accuracy | Humidity: +- 2% RH (Max +-5% RH)  Temperature: +- 0.5 °C |
| Resolution or Sensitivity | Humidity: 0.1% RH  Temperature: 0.1 °C |
| Repeatability | Humidity: +- 1% RH  Temperature: +- 0.2 °C |

Table 1 - RHT03 Technical Specifications

|  |  |
| --- | --- |
| Pin | Function |
| 1 | Power Supply |
| 2 | Data – Signal |
| 3 | NULL |
| 4 | Ground |

Table 2 - RHT03 Pin Sequence Number (1, 2, 3, 4 from left to right)

##### Communication and signal

Wire Bus is used for communication between the MCU and RHTY03.

**Illustration of MaxDetect’s wire bus:**

Data is comprised of integral and decimal part, the following is the formula for data.

* 8bit integral Relative Humidity data
* 8bit decimal Relative Humidity data
* 8bit integral Temperature data
* 8bit decimal Temperature data
* 8but checksum

When MCU send start signal, RHT03 change from standby-status to running-status. When MCU finishs sending the start signal, RHT03 will send response signal of 40-bit data that reflect the relative humidity and temperature to MCU. Without start signal from MCU, RHT03 will not give response signal to MCU. One start signal for one response data from RHT03 that reflect the relative humidity and temperature. RHT03 will change to standby status when data collecting finished if it don't receive start signal from MCU again.

Below one can find the overall communication process. The interval of whole process must beyond two seconds.

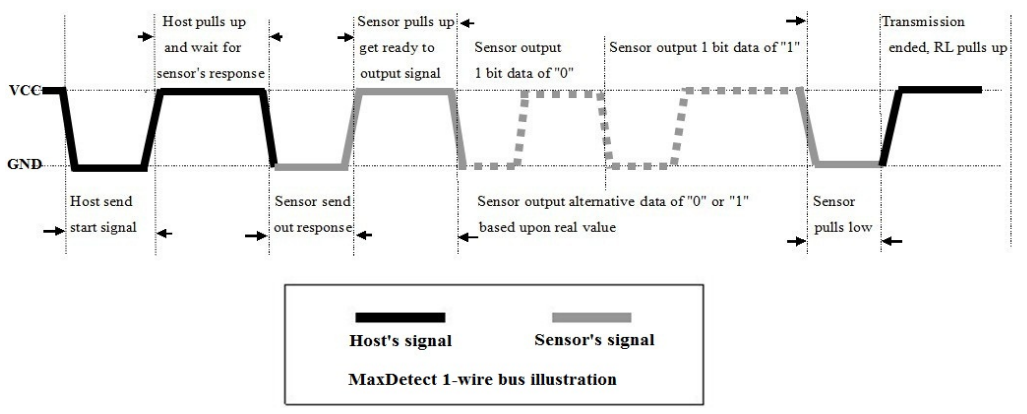


Figure 4 - RHT03 Communication Process

# References

1. Open Source Initiative: The Open Source Definition.

<http://opensource.org/docs/osd>

1. Open Source Hardware Association: OSHW Definition.

<http://www.oshwa.org/definition/>

1. Definition of Free Cultural Works: *Open Source Hardware.* [http://freedomdefined.org/OSHW](http://freedomdefined.org/OSHW%20)
2. Arduino: Homepage

<http://arduino.cc/en>

1. Arduino: *Reference.*

[http://arduino.cc/en/Reference/HomePage](http://arduino.cc/en/Reference/HomePage%20)

1. World Meteorological Organization, 1992: *International Meteorological Vocabulary*. Second edition, WMO-No. 182,Geneva.
2. Apache License, Version 2.0

<http://www.apache.org/licenses/LICENSE-2.0.html>

1. GNU General Public License, Version 2.0

<http://www.gnu.org/licenses/gpl-2.0.html>

1. System on a chip

<http://en.wikipedia.org/wiki/System_on_a_chip>

1. Google Charts API

<https://developers.google.com/chart/>

1. Arduino Mega 2560 R3 Web page.

<http://arduino.cc/en/Main/ArduinoBoardMega2560>

1. Arduino Ethernet Shield Web page.

<http://arduino.cc/en/Main/ArduinoEthernetShield>

1. Arduino Ethernet Library Web page.

<http://arduino.cc/en/reference/ethernet>

1. Arduino SD Library Web page.

<http://arduino.cc/en/Reference/SD>

1. RHT03 Official Datasheet.

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Weather/RHT03.pdf>