

## WEB Anemometer by Minas Kalarakis

### How a wireless home weather station turned into a web based weather station.

*An anemometer consists of two sensors, a wind speed and a wind direction sensor. Electrical Values from these two sensors are transmitted to a station where they are turned into human readable values.*

The meteorological services make regional weather forecasts for areas of broad interest, such as small ports, marinas, etc.

The r/c model flying Heraklion Crete Club is located in North Crete, 18Km from the city of Heraklion, 3000 meters away from the beach at an 800m altitude.

Initially, the club purchased a small weather station that included an anemometer, a thermometer, and a rain gauge. All items were depicted on the LCD screen. The signal was transmitted wirelessly from the sensors to the station

There was one drawback: to read about weather you had to be at the field or to call someone who was already there.

The Board has recently taken the decision to install a small solar power system (300w) to illuminate the field, provide access to the internet (wirelessly via a member of the club that his house is near the field), power a web Camera and a Skype phone. Now there was the opportunity to connect the weather station on the web

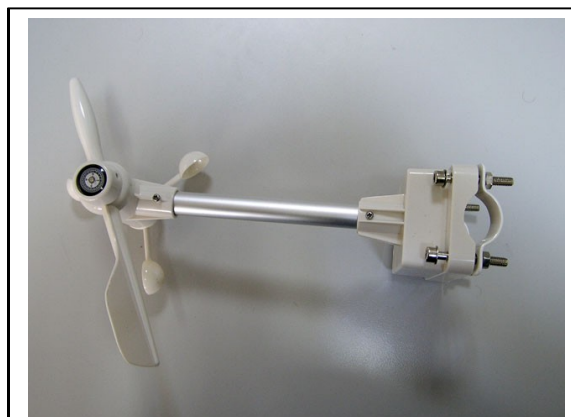
This article is focusing on how to isolate the wireless sensors and turn them into an independent web based weather station. Picture1.

The wind sensor was contained in a closed box with unknown electronic hardware and unknown way of measuring the speed and the direction of the wind.

Behind that, a cover reveals the electronic compartment. At first glance we could distinguish some parts that was the controller's pcb with a "black dot" in the middle, the wireless communication module, an NTC thermistor and a sensor for humidity, while there is a red led indicator that was flashing once per minute.

All of the above were removed and replaced with the PCB described below. Braids of 7 cables connect the sensors of speed and direction with the microcontroller board.

The disassembly of the two sensors was relatively easy; as simple as removing two screws from the side. The speed sensor consists of three cups which rotate within 5cm from the rotation axis. A magnet is mounted at the bottom and attracts a magnetic contact in each rotation. Wind speed can be calculated by the number of times the magnetic contact closes per second. This device gives an electric pulse of high-low.



Picture 1 The Anemometer which turned into a web device

The sensor of wind direction is a little more complex. It consists of two parts; the fixed part and a mobile indicator. The fixed part consists of a circular array of 8 infrared LEDs. In the center there are four receivers. The fixed part is notched so that the board has always the same orientation in the anemometer system. The four receivers in the center are oriented to the four cardinal directions. The station's installation instructions stated that the anemometer should be mounted so that the horizontal axis is against the north.

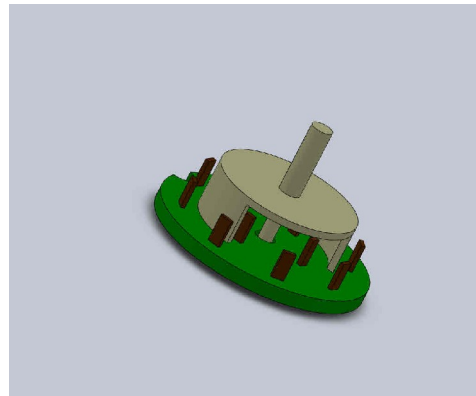
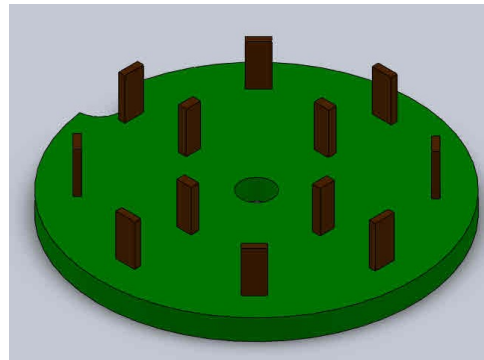
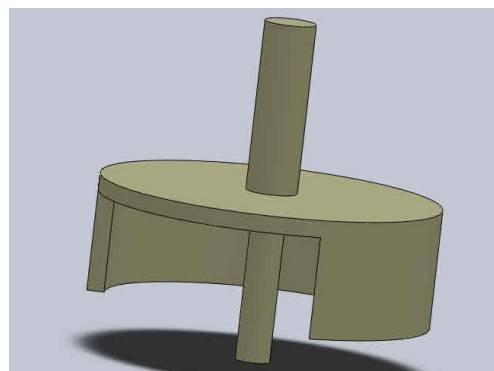
**Against of each receiver there is an infrared transmitter.** An infrared transmitter is positioned in every 45 degrees. In this way each receiver is illuminated by three transmitters. One against it one from 45 degrees at left and another from 45 degrees at right. Picture 3

The moving part consists of the arrow direction, attached to the inside cylinder which rotates between infrared transmitters and receivers in the center. On one side of the cylinder there is a window of 90 degrees. Thus when the cylinder is rotating at least one receiver is illuminated fully, while the two others receives little or none. Connecting them on the microcontroller A/D we are able to measure the change of voltage as receivers are illuminated and turn it into the wind direction through the program.

Another problem to be solved was that of the supply voltage. The primary sensor used two AA batteries for power supply. This led me to the conclusion that the supply voltage might be 3.3v; which was proved right.

The trial run with a LDO 3.3v. (TC2117-33) on a breadboard. By investigating the board I noticed that each receiver was connected to Vcc and the ground. It was fully powered. Each wire had a 3.3v voltage when it was unconnected. I connected a 10kiloohm resistor in **series to ground**. The common point was connected to an A/D on the microcontroller. When I rotated the cylinder I observed that at least one receiver was fully illuminated giving 2.5V. As the cylinder was rotating the receivers through the window was semi illuminated and either one of remaining receivers were obscured.

Eventually when the direction changed more than 90 degrees, the receiver was not illuminated and had a voltage below 200milivolt. Simultaneously the receivers towards



Picture 3 A graphic, describing the direction sensor

the direction of rotation will increase their voltage. The central one had a voltage of 2.5volts and the other two ranged from 200milivolt to 2.5volts.

Finally, by observing the circuit design I came to the following conclusions;

The supply voltage was at 3.3v (or at least it worked fine with that).

The four led receivers can be connected to the A/D on the microprocessor.

Voltage combinations correspond to the wind directions.

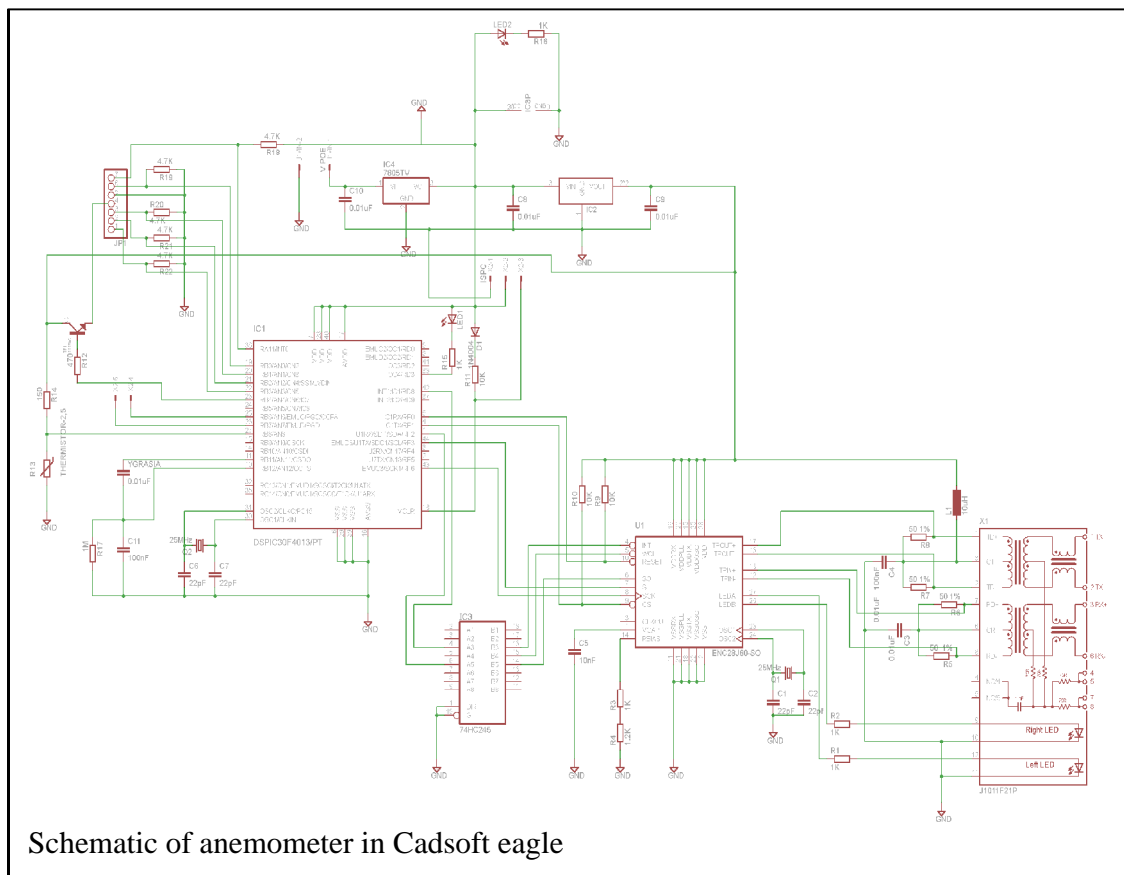
The whole device is made in such a way to be completely weather resistant.

I put a label on each wire describing its function and I reassembled the sensor.

## Replacing the main board

The processor I used in this application is a Microchip dsPic40F13, clocked by an 8MHz external crystal. The microprocessors PLL is adjusted by a factor of 8X, setting the internal operating frequency at 64 Megahertz. A clock greater then 40 Mhz is recommended to get from 8 to 10 Mhz SPI clock

The speed of the wind, as mentioned above, is measured by the speed of a pivot magnet that passes in front of the magnetic contact. The contact is linked to a resistance



of 10kilohm. The magnetic contact is connected to ground. The common point is connected to RA11/INT0. The INT0 routine is activated so that every time the magnetic contact closes it causes an interrupt to the microprocessor.

The wind direction consists of four inferred receivers which are connected to A/D channels on microcontroller. Voltage values range from 200miliovts when not

illuminated to 2.5volts when illuminated fully. The program can turn them into wind direction.

Inside the original anemometer there was a 10Kilohm (at 20 C) NTC thermistor for temperature and a humidity sensor. The thermistor is used for the temperature sensor via a 10kilohm resistor connected to pin 27 RD8/AN8

The humidity sensor was not implemented in this project. Initially a DHT-11 Digital Temperature and Humidity Sensor was planed to be used. It is a 3.3v one wire device. The sensor was not delivered on time and was left aside for future implementation.

A ENC28J60 chip was used for communication to the internet. The ENC28J60 chip is a stand-alone Ethernet controller with an industry standard Serial Peripheral Interface (SPI).

Communication with the host controller is implemented via the SPI, with clock rates up to 20 MHz.

The ENC28J60 is a 3.3V powered module. However, it was designed to be easily integrated into 5V systems. The SPI CS, SCK and SI inputs, as well as the RESET pin, are all 5V tolerant.

An economical 74HCT08 (quad AND gate), or 74HC245 (octal 3-state buffer) or many other 5V CMOS chips with TTL level input buffers may be used to provide the necessary level shifting. Some 74HC245 chips in TSSOP20 package were in my drawer, and one of them provide the level shift from ENC28J60 to Microchip dsPic33F4013. The pins INT and WOL are level shifted to controller but they aren't implemented in the software.

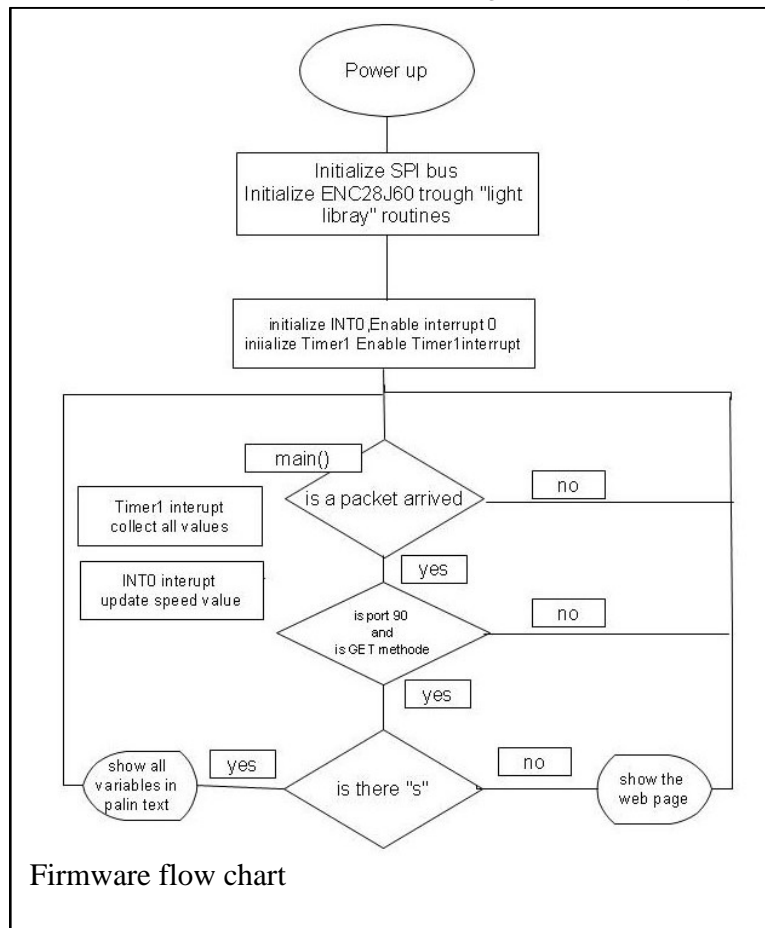
Two dedicated pins are used for LED link and network activity indication. The LEDA

and LEDB pins support automatic polarity detection on Reset.

The ENC28J60 is designed to operate at 25 MHz

A typical application and schematic can be found inside the ENC28J60 manual.

To complete the Ethernet interface, the ENC28J60 requires several standard components to be installed externally.



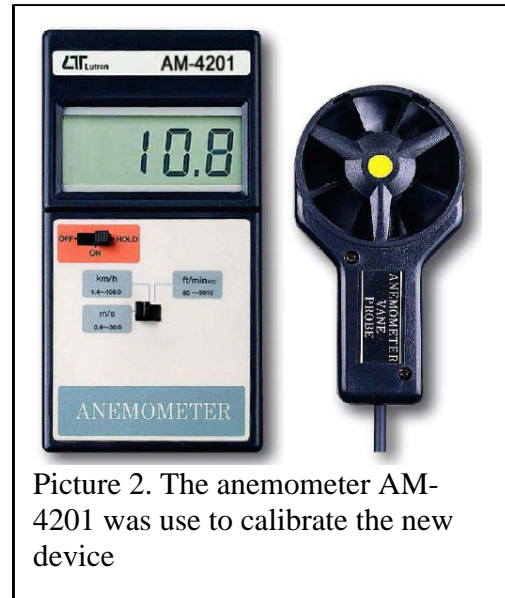
For the TPIN+/TPIN- and TPOUT+/TPOUT- pins 1:1, centre taped pulse transformers (rated for Ethernet operations) are required. When the Ethernet module is enabled, current is continuously sunk through both TPOUT pins. When the PHY is actively transmitting, a differential voltage is created on the Ethernet cable by varying the relative current sunk by TPOUT+ as compared to TPOUT-.

The Ethernet transformer module J1011F21P from Pulse electronics meets all the specifications required to connect a cat5e cable to ENC28J60 transmission lines.

Both transmitter and receiver interfaces additionally require two resistors (50 ohm 1%) and a capacitor (100nF) to properly terminate the transmission line, minimizing signal reflections. Two leds (green and yellow) are also integrated on the module to indicate Ethernet activity.

The whole circuit requires less than 500 milliamps, of which 100milliamperes pass through the wind direction sensor.

The 5V voltage is supplied by a LM7805 chip. A TC2117-33 LDO steps down the voltage to 3.3v for the ENC28J60. All other components on the PCB require a 5v power supply



Picture 2. The anemometer AM-4201 was used to calibrate the new device

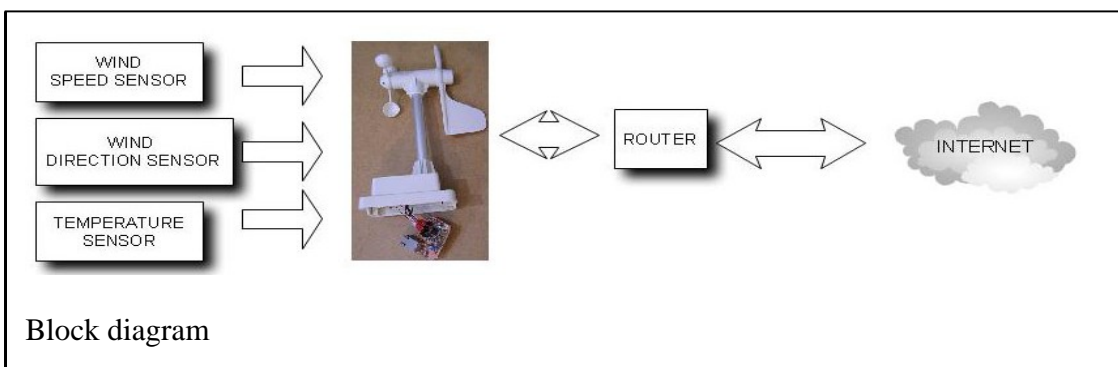
## Writing firmware.

The main requirement is to access the anemometer remotely. This feature provides the Ethernet protocol and ultimately the World Wide Web.

The programming language is C – Mikroelektronika C. The total code is about 10Kilobyte including the web page written in plain HTML.

Mikroelektronika C is equipped with a library for controlling the ENC28J60. There is a light and an extensive version of the library. A detailed example of the use of the light version can be found in Mikroelektronika C's help file. The code can be embedded implicitly in the project's code. Most of the development time was spent in collecting and processing the data from the sensors. Management and low level programming of the ENC28J60 took the lightweight version of the library in MikroC

The speed sensor is connected to INT0 and brings about an interrupt each time the magnet on the rotating part passes by the magnetic contact. In such way the program does not measure how long it takes a rotation but how many revolutions does the sensor per second (global variable AirSpeedTick). The only action in routine INT0 is to increase the counter- variable Airspeedtick - which measures how many times slopes of the



magnetic contact have

Time monitoring has been assigned to timer1, which beats once per 250milisecond. In every second the Tmer1 interrupt routine dose the following

- a) The pin RD3 is reversed. A red led is connected on it; it blinks indicating that the circuit is in normal operation.
  - b) Increase the counter which counts how many milliseconds have passed. When it reaches 4 (one sec), it fires the procedure to convert the variable AirSpeedTick into wind speed in meters per second and then it converts it to kilometers per hour. The variable AriSpeed is updated, which will be displayed on the website. It calculates the maximum and minimum wind speed and updates the appropriate variables
- NTC thermistor is connected on A/D channel 8, The value is red and stored to global variable NumTemp

To adjust the wind speed sensor the anemometer AM-4201 was used. Picture 2. During measurements I observed that the speed of the wind is proportional to the number of spins multiplied by a factor of 9. In the same routine the average maximum and minimum of wind speed are also calculated.

The same routine is responsible for the wind direction. Because the infrared receivers are connected to the microprocessor A/D, it is possible to calculate the wind direction accurately. In practice there is no need for greater accurecy it is sufficient for the clue N or N-NW- S.

Because of this, before calculating the wind direction from the values of infrared receivers' voltage, these values are passed through the routine NormalizeValues () in which all values greater or equal 2000 are set to 2000, values from 1000 to 1999 are set to 1000, from 300 to 999 are set to all 500, while values smaller than 299 all become 200. By doing that the

routine SetAirDirection () ends up with 17 wind directions. Changing value limits flattened routine **NormalizeValues ()** may further increase the accuracy in wind direction. e.g. the range from 300 to 999 which are all 500, can be divided into two new parts e.g. Values from 300 to 500 to be all 400 and from 501 to 999 to be all 600. Microprocessor's A/D channels 0 -3 are attached to the four infrared sensors. Each receiver, depending on the illumination reception yields values from 200milivolt, when not illuminated, to 2500milivolt when illuminated fully. Normalizevalues () limits the range of these values. The new values are stored in the variables ALeds []. Returning

AN0	AN1	AN2	AN3	DIRECT
500	200	500	2000	N
2000	200	200	2000	N-NW
1000	200	500	2000	N-NW
2000	200	500	2000	NW
2000	500	500	1000	NW-W
2000	500	500	500	W
2000	1000	200	500	W-SW
2000	2000	500	500	SW
1000	2000	500	200	SW-S
2000	2000	500	200	S
500	2000	1000	200	S-SE
500	2000	200	500)	SE
200	1000	2000	500	SE-E
200	500	2000	500)	E
500	500	2000	1000	E-NE
200	200	2000	2000	NE
500	200	1000	2000	NE-N

Table 1 : values from A/D are flatten

values from NormilizeValues () in Table Aleds[] can produce 17 combinations. The routine SetAirDirection () converts these combinations to cardinal points of wind direction (N - NE-E etc) see table 1

The main() function contains an endless loop with one function, the SPI\_Ethernet\_doPacket() function which looks for arrived packets. If a packet arrives all appropriate variables hold fresh values and they are ready to be transmitted through Ethernet. The Anemometer listens at port 90. The source code as copied from the paradigm supports only the GET method. The getRequest variable holds the number of requests served so far. If a request's path name starts with "s"

([www.eahk.dyndns.tv:90/s](http://www.eahk.dyndns.tv:90/s)), the anemometer transmits plain text with all value and pure data got from the A/D. This text consists of A/D vales, normalized values, Airtick variable which is the times per second the magnetic contact slopes, temperature value and the number of requests which had been served so far.

This mode enables users who want to gather data for long periods. If there is nothing on the left of the URL it transmits the main page. The main web page which anemometer replies to users is written in plain HTL language, which is a standard for all web browsers.

When a TCP packet arrives "light library", routine SPI\_Ethernet\_doPacket calls the function SPI\_Ethernet\_UserTCP() The reply HTML page is a part of this routine.

According to the router's configuration, the static IP 192.168.0.70 and gateway 192.168.0.1 were assigned to the anemometer. As it is a device inside the local network, it is hidden behind the router; the MAC address used in the paradigm was assigned.

```
void onTimer1Interrupt() org 0x00001A {
    int i;
    IFS0bits.T1IF = 0;           // clear interrupt flag
    PORTDbits.RD3 = ~PORTDbits.RD3 ;
    TickCounter++;
    TempTick++;

    asm CLRWDT;           // clear WDT Timer

    //every 4 tick read values and then a) normalize them , put
    //appropriate values to arguments for the web page
    if (TickCounter>4){
        ReadValues();
        NormalizeValues();
        SetAirDirection();

        AirSpeed=AirSpeedTick * 10 /AirSpeedCoficient; // 9 turns/sec
        = 10Km/H
        // alter max and min air speed acording previousAirSpeed
        if (AirSpeed >PreviousAirSpeed) {maxAirSpeed=(maxAirSpeed +
        AirSpeed) /2;}
        if (AirSpeed < PreviousAirSpeed) {MinAirSpeed = (MinAirSpeed
        + AirSpeed) /2;}

        if (AirSpeed >maxAirSpeed) {maxAirSpeed=AirSpeed;}
        if (AirSpeed < MinAirSpeed) {MinAirSpeed = AirSpeed ;}

        AverageAirSpeed=(PreviousAirSpeed + AirSpeed)/2;
        if (AverageAirSpeed==0) {maxAirSpeed=maxAirSpeed /2;}
        PreviousAirSpeed=AirSpeed;

        integTemp=ADC1_Read(8) /59;           // read Temperature

        if (TempTick>240){           // one minute passed
            // shift right 14 previous values
            for (i=0;i<14;i++){
                LastQuarterSpeed[i]= LastQuarterSpeed[i+1];
            }
            LastQuarterSpeed[14]= AverageAirSpeed;
            TempTick=0;
        }
        AirSpeedTick=0;
        TickCounter=0;
    }
}
```

Timer1 interrupt routine. All begins here.

From bench to the real world.



The anemometer case is designed to withstand all weather conditions. It is sealed and water proof. In the front, there is a compartment for two AA batteries. Behind that, a cover reveals the electronic compartment. The case is 40mm wide. The new PCB should be no more than 40mm X 60mm in size to fit inside the anemometer's case. Thus the chips were soldered on both sides of the PCB.

A 30 meter cat5 cable is used to connect the router and the anemometer. The ENC28J60 chip is an IEEE 802.3™ Compatible Ethernet Controller with 10Base-T PHY that requires two pairs from 4 pairs there are inside the cat5 cable.

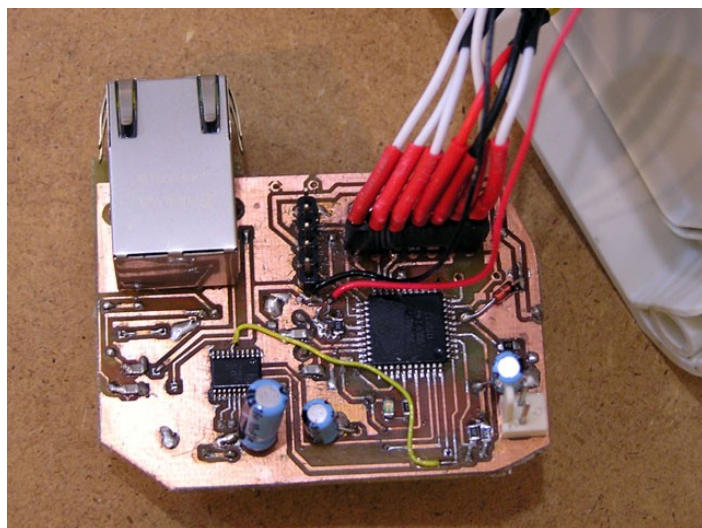
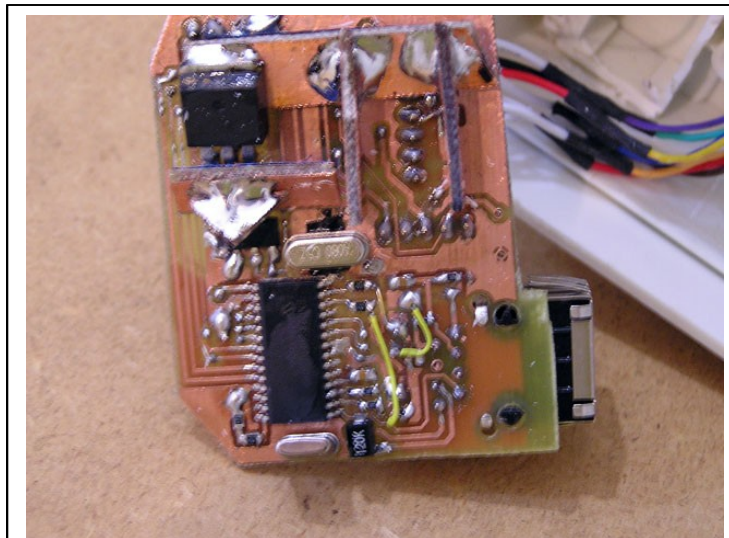
Blue and brown pairs were used for power. The blue pair was connected to the positive pole through a 1 ampere fuse and an on-off switch. The brown pair was connected to the negative pole of the battery.

The Switch was turned on and anemometer went live!

The red led connected to RD3 begins to flash indicating that the board was functioning properly.

Over this led, on the cover, there is a hole. The user can see the red led flashing and warm air can exit the case. There are small holes at the bottom of the case. . Summer days in CRETE are very warm - temperature easily climbs over 40 C- and the device is going to work perpetually.

Using my laptop I entered the url : [www.eahk.dyndns.tv:90](http://www.eahk.dyndns.tv:90) at the browser address bar and I got the first live measure as picture 5. Android phones could also display the page well. Some of the club members had iphones with safari browser. The phones displayed the html page but Greek letters were not shown correctly. The character set on the HTML is iso-8859-7, while it had to be UTF-8.



Picture 4. The new pcb must be 40X60mm Parts are soldered on both sides



Another problem reported later at night was that the HTML page had to be narrower, 400 pixels wide. Some members wanted to embed the anemometer web page to their web blogs. Also, they would like all labels to be translated in English. Real wind speed measurements show that a small correction must be done in the speed coefficient in order to be more accurate. Members want to know speed history for last quarter. That numbers are shown at the bottom of the table. The firmware is already updated in anemometer covering all members' suggestions.

[Reload](#)

TEMERATURE	WIND	DIRECTION	HUM
19 C	max 8	W	---- %
	6 Km/H		
	min 2		

5
6 6 8 6 2 2 4 4 6 7 6 8 8 6

[@Kalarakis.gr](#)

Picture 5. Anemometer's main web page



This document was created with Win2PDF available at <http://www.win2pdf.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.  
This page will not be added after purchasing Win2PDF.