## http://upload.wikimedia.org/wikipedia/commons/thumb/9/90/Quadrocopter_Multicopter_DJI-S800_on-air_credit_Alexander_Glinz.jpg/300px-Quadrocopter_Multicopter_DJI-S800_on-air_credit_Alexander_Glinz.jpg7.2. TERRAIN TEMPERATURE MAPPING

The second application developed was a device which could carry out a thermal terrain mapping, designed in order to be able to be installed in most standard professional civil drones. This device is intended to offer a much cheaper alternative to the conventional terrain mapping by using low cost temperature sensors instead of thermal cameras, which have a relatively high cost (see section 5.2.2.3. Other Portable Devices), and mounting them on drones rather than on helicopters, reducing the operation cost.

Figure 53: Photograph of a standard professional civil drone with a camera as payload.

Thermal terrain mapping is used for several purposes, such as crop irrigation control for plantations, vegetation monitoring, fire monitoring (both subterranean and surface fires), high voltage lines monitoring, thermal efficiency of buildings, search of lost people in inhospitable regions, among others.

In order to carry out a temperature mapping of an area, the device relies on the Melexis 90614-ACF IR sensor (previously studied in section 6.4.2. ASSAY OF THE INFRARED SENSOR) to measure the surface temperature of the terrain while the device is mounted on a flying drone. Unlike thermal cameras, which do an instantaneous thermal photograph of an area, the Melexis IR sensor takes punctual measures of the surface’s temperature inside its field of view which makes it unable to do a thermal terrain mapping by itself. In order to work this problem around, every 0,2 seconds the device registers a temperature measurement and the position where it was taken obtained from a GPS installed in the device. By flying the drone all around the terrain that is wanted to be analyzed, a full thermal terrain mapping can be obtained. All the data taken is saved into an SD card using an SD logging device, installed in the device.

In order to carry out these operations, the device had the following requirements:

* Capacity to determine its position using a GPS.
* SD card data logging capabilities.
* Power autonomy.

### 7.2.1. MATERIAL

For this application, the material that was used to form the module will be presented first for convenience.

In order to cover the GPS position and the SD card data logging requirements, the Adafruit Ultimate GPS data logging shield was obtained [33]. This Arduino shield has installed a MTK3339 GPS (a -165dBm, 66 channels, 10Hz update GPS chipset), as well as an SD card logger which comes with an included level shifter. It also has some proto-shield space, which makes possible to solder circuits on the shield itself.

The GPS has an antenna included, so no external antenna is needed. However, it has a connector in case one is wanted to be used if the module is intended to be placed in a confined space with no clear view of the sky. It was considered that for this application an external antenna was not necessary.

The SD card logger works as any other SPI-communication based SD card logger would, and the internal connections of the shield are equivalent to the ones presented before in section 7.1.4.1. Wiring for data logging with SD cards.

The IR sensor used, as mentioned before, is the Melexis 90614-ACF IR sensor. In order to measure the terrain’s temperature this sensor needs to be put upside down. This carries along a difficulty, as it cannot be mounted on the proto-shield space because the GPS needs to be pointing upwards. This made necessary to make an additional small platform to hold it in place.

This module was firstly planned for an Arduino UNO. However, even though all the software and wiring would be correct, the program would not work as expected. Finally, after corroborating that the code was actually correct, it was determined that the cause of the problem was a memory issue. As explained in section 6.1.1.1. ATmega328 Microcontroller Specifications and Memory, the Stack used memory grows from top to bottom and the Heap from the Static Data upwards. If both two memories grow to much they finally collide and one overwrites the other, causing program malfunction. As a first solution, the code was optimized and reduced, but the problem was not solved. In order to be able to further reduce the memory used by the program, it should have been rewritten without using libraries and using low-level instructions. However, due to time restrictions, this was discarded as an option.

Instead, an Arduino MEGA was obtained and the code was adapted for this platform. The Arduino MEGA runs on a different microcontroller of the same family (which make programs designed for the Arduino UNO highly compatible with the MEGA), the ATmega1280. It is designed as a platform which is able to support a higher amount of inputs and outputs, as well as more serial communications. However, the reason why it was picked (as this module is actually working with a small amount of external devices) is because it has more memory available (see Table 6). With this increase in memory, the program was able to run perfectly.

The overall cost of the whole module was of 36,89€. The price breakdown can be found at Table 8.

|  |  |
| --- | --- |
| Item | Cost |
| Melexis 90614-ACF IR sensor | 17,45€ |
| Adafruit Ultimate GPS data logging shield | 40,87€ |
| Arduino MEGA | 35€ |
| Other (wires, drilled plate, resistances, etc) | 3€ |
| Total | 96,32€ |

Table 9: Cost breakdown of the termal terrain mapping module for drones.

### 7.2.2. GPS POSITIONING

In this section, the operation with the MTK3339 GPS through the Adafruit Ultimate GPS data logging shield will be explained. The SD data logging will not be explained as it is completely equivalent to the one explained in section 7.1.4. DATA LOGGING.

#### 7.2.2.1. Wiring of the shield with the Arduino MEGA

Due to the fact that the shield is designed for Arduino UNO and an Arduino Mega was used, the shield had to be adapted for its operation with the MEGA. The main problem is that serial software is used on pins 7 and 8 for communication with the GPS when using an UNO, but software serial is not supported by the Arduino MEGA. Along with some code, in order to work around this problem, the Tx and Rx (transmit and receive) pins of the GPS needs to be bridged and connected to the Rx and Tx of a Serial communication of the Arduino MEGA (Arduino MEGA has 3 independent serial communication channels).

#### 7.2.2.2. Software to interface the shield with the Arduino MEGA

In this section the code needed in order to make the GPS work is explained. This lines of code are the used in this module’s program.

##### 7.2.2.2.1. Libraries

* **Adafruit\_GPS.h:** This library developed by Adafruit gives the user a straighter way of asking the GPS for information.
* **SoftwareSerial.h:** This library is needed for the serial communication between the Arduino and the GPS.

##### 7.2.2.2.2.Global variables

* **Hardware Serial:** a hardware serial is started. It is defined through which serial ports it will be held. In the case of using an UNO, this would be a Software serial through pins 7 and 8.
* **Adafruit\_GPS *GPS*(&*mySerial*):** The GPS object is created. In this line it is also related to the hardware serial defined before (*mySerial*), which will be the channel used to communicate.
* **GPSECHO:** this Boolean can be activated or deactivated in case that an echo of the data received by the GPS is wanted to be displayed to the user, for debugging purposes.
* **Boolean usingInterrupt:** this Boolean activates or deactivates the use of an interrupt which will e explained later.

##### 7.2.2.2.3. Setup

In the Setup, the serial communication is started, the SD card initialized and the file where the data is going to be saved created.

* **Serial.begin(115200):** This command initializes the serial port for terminal-communication with the PC, useful for debugging. It is started at the 115200 baund because it permits a higher frequency of measurements and because the 9600 baund will be occupied by the Arduino-GPS communication.
* **GPS.begin(9600):** Arduino-GPS communication is started through the 9600 baund serial communication.
* **GPS.sendCommand(*option*):** This instruction sends set up instructions to the GPS.If *option* is PMTK\_SET\_NMEA\_OUTPUT\_*MODE*, it is being communicated to the GPS which kind of data is wanted by changing *MODE* to different options; with PMTK\_SET\_NMEA\_UPDATE\_*FREQUENCY* the rate at which the GPS data is updated is set to the frequency defined by *FREQUENCY*; finally if PGCMD\_*ANTENNASTATUS* is sent, it is being said to the GPS if an external antenna is being used or not through the *ANTENNASTATUS* line.

##### 7.2.2.2.3. Functions and interrupts

In order to use the GPS it is recommended the use of an interrupt which is called once a millisecond, looks for any new GPS data and stores it.

##### 7.2.2.2.4. Loop

In the loop, first it is checked if new information is received through the **GPS.newNMEArecieved()** instruction. The GPS data is transmitted to the Arduino through the **GPS.parse(GPS.lastNMEA())** instruction. After parsing it, the information can be accessed through instructions which come from the Adafruit library, such as GPS.hour, GPS.minute and GPS.seconds, which are time information; GPS.fix and GPS.fixquality which are satellite-connexion information; GPS.latitude and GPS.longitude, which gives the location; among others.

### 7.2.3. WIRING

In this section the wiring for the module will be explained. As a shield is being used it is a really simple operation:

* The shield is plugged onto the Arduino MEGA. This sets up all the connections with the SD card and the GPS.
* The serial bridge explained at section 7.2.2.1. Wiring of the shield with the Arduino MEGA is done (in the module it was connected to the Arduino MEGA Serial1).
* Due to the fact that Arduino MEGA has its own SPI dedicated pins, the Melexis sensor does not need to be plugged to the analog pins, but it is plugged to the SDA and SCL pins (pin 20 and 21). Power and ground are also connected.

### 7.2.4. SOFTWARE

For this application, the device’s Arduino code as well as the post-process code will be explained.

#### 7.2.4.1. Arduino Code

The code for Arduino is a combination of the codes for the Melexis IR sensor which are explained at section 6.3. SOFTWARE FOR INTERFACING THE ARDUINO AND THE SENSORS, adding the code necessary for data logging into an SD card explained previously at section 7.1.4.2. Software for data logging with SD cards and the one necessary for communication with the GPS previously seen at section 7.2.2.2. Software to interface the shield with the Arduino MEGA. The full code can be found at the annex section 1.4.1. ARDUINO CODE.

The code works as follows:

* The libraries and global variables are initialized.
* In the setup, the sensor, the SD card and the GPS are initialized. The SD logging file is created. The type of information that is wanted is sent to the GPS, as well as its frequency (5Hz for the application).
* Once every millisecond, an interrupt will force the actualization of the GPS’s data.
* In the loop, if new position data is received from the GPS, it is parsed. If there is a fix (the GPS received Satellite data), the GPS position information and the temperature registered by the Melexis are saved into the log file in the SD card.

#### 7.2.4.2. MatLab Code

The MatLab code can be found at the annex section 1.4.2. MATLAB CODE (POST-PROCESS). The code imports the data from the log txt file and assigns the longitude, latitude and the temperature to different vectors. Then, a 100x200 mesh is created using the **meshgrid** instruction. The mesh divisions should be modified as function of the area scanned. With the MatLab’s griddata function, the experimental data is plotted on the grid and from this data, the grid’s node data are calculated. With this system, instead of having punctual values of temperature, the value of temperature at the nodes of the grid is approximated and then a linear approach between nodes is carried away, giving a continuous temperature measurement. Finally the results are plotted. This plot can be exported to jpg and then be added as a layer on Google earth, giving a temperature mapping over the satellite photograph of the terrain.

### 7.2.5. TEST IN A REAL ENVIRONMENT

Due to lack of time, the module could not be tested in its planned real environment (mounted on a drone). However, before-flight tests were carried away, which had similar conditions. This tests were carried away in order to debug the code and look for errors before mounting the device on a drone, as when mounted debugging becomes much slower.

NOTE: As per this revision date (3/4/2015) this module was actually tested mounted on a drone on 06/01/2015 at the R/C-model aircraft airfield “Els Cards”. The data used as example for the post-process software is from that session.

In order to recreate a real environment, the device was attached to the end of a telescopic stick. This would recreate the fact of flying, enabling to see the measurements acquired by the Melexis IR sensor from an approximately 3 meters height from the ground. The device was not fully autonomous; as this activity was carried away for debugging thus the Arduino was connected to a PC through a long USB cable.

#### 7.2.5.1. Test site

The test was carried away at 1:15pm on 23rd December 2014 at Esplugues’ Pompeu Fabra Park (C/Professor Barraquer). This site was decided for its vicinity and variable terrain materials (sand, concrete, plants, …). As the sun was low due to winter time, the shadows from the buildings south to the park generated two temperature areas in the park too, due to the fact that the day before had rained and the side which was under the shadow did not dry. In order to see if the device was capable to “see” relatively small hot spots, three people sat on the floor at different positions in order to check if the IR sensor measurement changed.

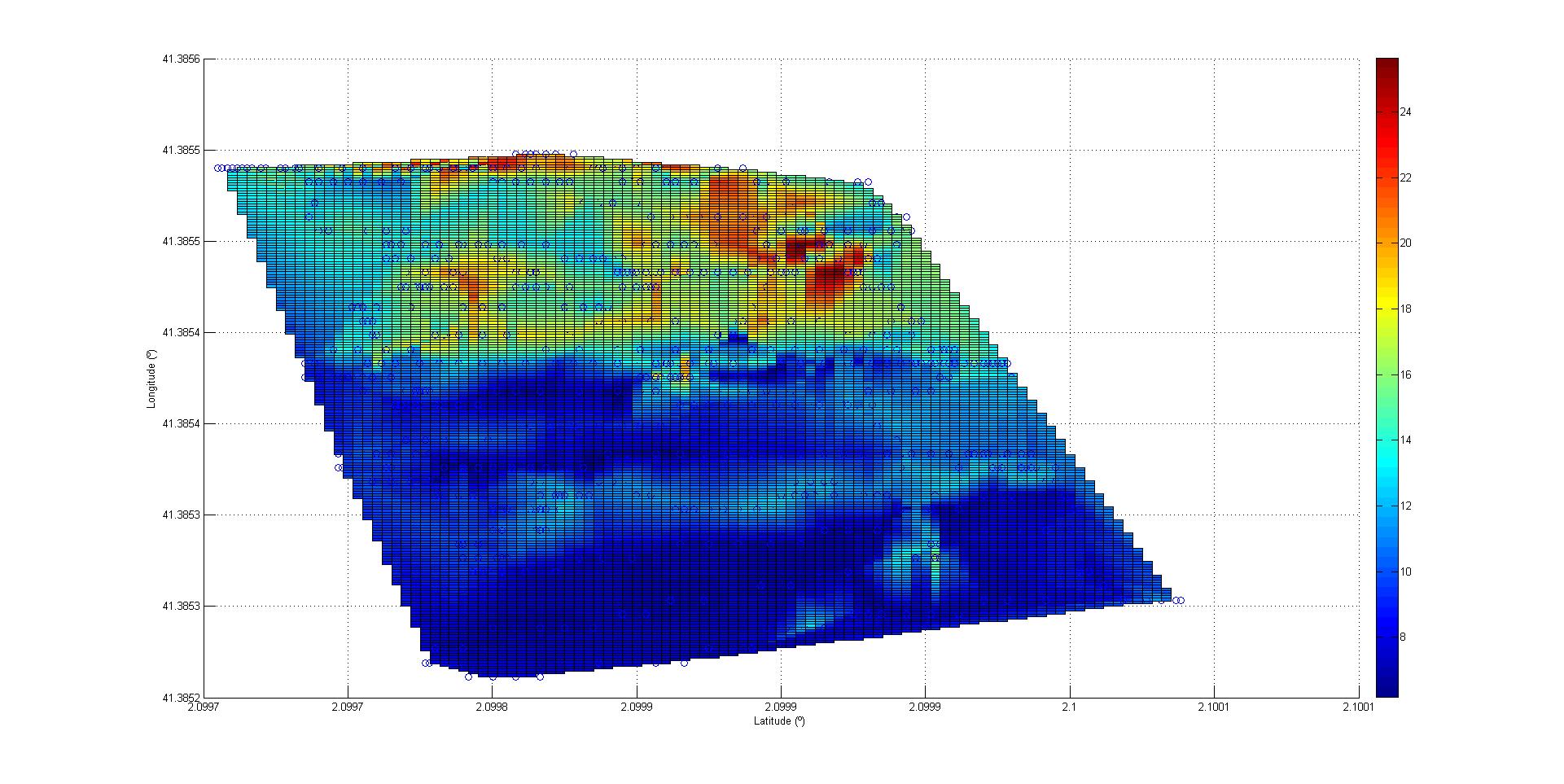
|  |  |
| --- | --- |
| C:\Users\Carles\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Untitled-3.jpg | Untitled-2 |

Figure 54: Photograph of the test site (left) from Google Earth and diagram (right) showing the shadowed area (blue), the sunlit area (red) and the hot spots (yellow).

#### 7.2.5.2. Results and analysis

The data obtained in the test can be found at the annex section 3.2. TERRAIN TEMPERATURE MAPPING. The data was post processed with the MatLab code explained at section 7.2.4.2. MatLab Code. The plotted results can be seen at Figure 55.

It can be observed that the upper half of the graph is overall hotter than the lower half. This is due to the fact that, as explained in section 7.2.5.1. Test site the lower part of the terrain was shadowed while the upper one was sunlit.



**Figure 55**: **Graph displaying the plotted results for the Thermal terrain mapping for drones test. Color scaling represents temperature readings. In dark blue both the grid and the data points can be seen.**

In Figure 56 the resulting plot is superimposed to the Google Earth’s satellite image of the terrain scanned. With this image details of the results will be further analyzed. It is worth noting that the MatLab’s plot does not have a good scaling between the longitude and the latitude, which forces to adjust the image so that it fits well with the actual area scanned.

On the lower part of the cold area two hot spots are visible where two people were sitting. The spot’s temperature is about 8ºC higher (around 16ºC). This is not a human-body temperature but this is due to the fact that the temperature measured is not the person’s but the clothes’ he was wearing. It can be seen that the upper right spot is bigger. This is because there was more density of measurements in that region than in the lower left spot. The third hot spot can be seen in the limit between the hot region and the cold region, in the middle of the map, as a 20 to 22ºC spot.

It also can be observed that there is an horizontal line of hotter measurements across the cold area. These hotter readings are caused by the presence of vegetation, which retains the temperature better than the ground. This fact can be also observed in the hotter region, where those plants could reach up to 25ºC.

It must be mentioned that as it can be seen in the image, some vegetated areas are not hotter. This is caused by the fact that the maximum altitude obtainable by the telescopic stick was about 3m height and that vegetated areas are trees. On this areas the sensor scanned the area under the tree’s crown as it was impossible to do it otherwise, so it is actually the ground temperature, not the trees’.

Although not being visible neither in the graph nor on the image, there is a cold spot in the middle of the terrain near the hot spot forced by the presence of the person. This cold spot is the measurement of temperature of a toboggan slide, which was made of polished metal. The temperature measured in this zone was even under zero at some points. Rather than being the actual temperature of the slide, this measurement is caused because of the high emissivity of polished metals, which along with the set up emissivity of the Melexis IR sensor causes these low measurements.

Overall the results are satisfactory and could be useful for different applications, although they are highly dependent of data concentration and, as seen by the toboggan slide, of emissivity. Even if the height at which it was tested was not as high as if mounted on a drone, it is believed that such good results will repeat.

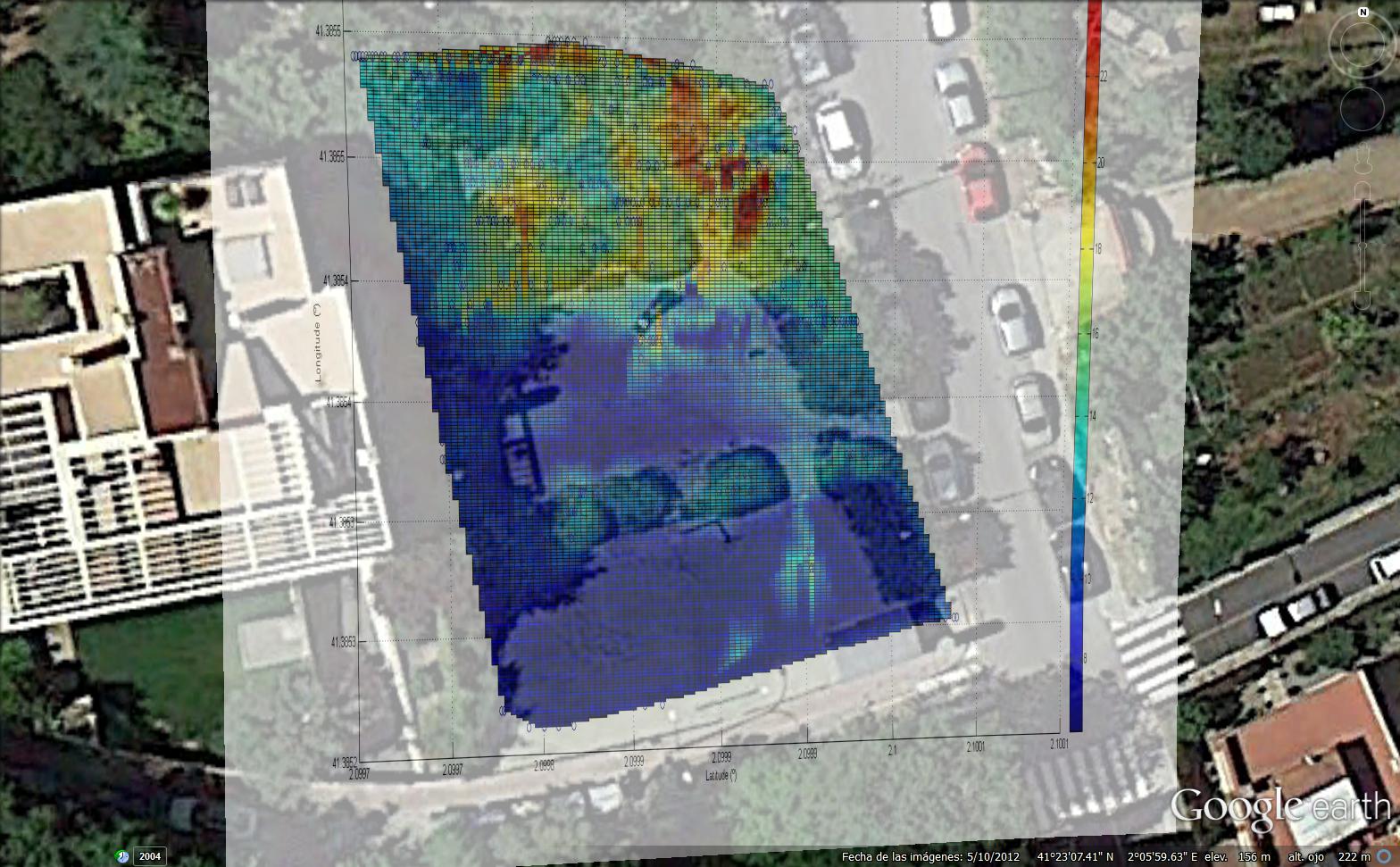


Figure 56: Image of the terrain termal mapping test superimposed in Google Earth over a satellite image of the terrain.

# 10. BIBLIOGRAPHY

|  |  |
| --- | --- |
| [1] | T.P.Wang, "Thermocouple Materials," *ASM - The Materials Information Society,* pp. k88 -k107, 1990. |
| [2] | Maxim Integrated, "Implementing Cold-Junction Compensation in Thermocouple Applications," 26 April 2007. [Online]. Available: http://www.maximintegrated.com/en/app-notes/index.mvp/id/4026. [Accessed 03 November 2014]. |
| [3] | National Instruments, "Cómo Realizar una Medición con Termopares," 19 August 2013. [Online]. Available: http://www.ni.com/white-paper/7108/es/. [Accessed 02 November 2014]. |
| [4] | A. Tong, "Improving The Accuarcy of Temperature Measurements," 2001. [Online]. Available: http://www.picotech.com/applications/temperature.html. [Accessed 2014 November 03]. |
| [5] | National Instruments, "Guía para Realizar Medidas de Temperatura con RTDs," 19 August 2013. [Online]. Available: http://www.ni.com/white-paper/7115/es/. [Accessed 03 Noveber 2014]. |
| [6] | Agilent Technologies, "Practical Temperature Measurements," 26 January 2012. [Online]. Available: http://cp.literature.agilent.com/litweb/pdf/5965-7822E.pdf. [Accessed 03 November 2014]. |
| [7] | M. Massoud, Engineering Thermofluids: Thermodynamics, Fluid Mechanics, and Heat Transfer, Berlin, Germany: Springer, 2005. |
| [8] | Scigiene Corporation, "Infrared Thermometers," [Online]. Available: http://www.scigiene.com/pdfs/Infrared%20Thermometers.pdf. [Accessed 03 November 2014]. |
| [9] | OMEGA, "Introduction to Infrared Thermometer," [Online]. Available: http://www.omega.com/prodinfo/infraredthermometer.html. [Accessed 03 November 2014]. |
| [10] | OMEGA, "Digital Thermometer HH11B," [Online]. Available: http://www.omega.com/pptst/HH11B.html. [Accessed 02 December 2014]. |
| [11] | OMEGA, "RDXL-SD Series Portable Thermometer/Data Loggers with SD Card and Thermocouple Input," [Online]. Available: http://www.omega.com/pptst/RDXL-SD\_SERIES.html. [Accessed 02 December 2014]. |
| [12] | OMEGA, "Compact Portale Data Logger," [Online]. Available: http://www.omega.com/pptst/RDXL120.html. [Accessed 02 December 2014]. |
| [13] | AllQA, "AllQA Infrared Thermometers," [Online]. Available: http://www.allqa.com/infrared.html. [Accessed 06 December 2014]. |
| [14] | Fluke, "Fluke VT04 Visual IR Thermometer," [Online]. Available: http://www.fluke.com/fluke/m3en/thermometers/infrared-thermometers/fluke-vt04.htm?PID=77085. [Accessed 06 December 2014]. |
| [15] | National Instruments, "Dispositivo de Medidas de termopares NI USB-TC01," [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/es/nid/208177. [Accessed 07 December 2014]. |
| [16] | National Instruments, "Sistema de Medidas NI 9215," [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/es/nid/209871. [Accessed 07 December 2014]. |
| [17] | Arduino, "Arduino UNO," [Online]. Available: http://arduino.cc/en/Main/ArduinoBoardUno. [Accessed 7 December 2014]. |
| [18] | B. Earl, "Learn Adafruit: Arduino Memories," Adafruit, [Online]. Available: https://learn.adafruit.com/memories-of-an-arduino/arduino-memories. [Accessed 07 December 2014]. |
| [19] | Arduino, "Arduino Software," [Online]. Available: http://arduino.cc/en/Main/Software. [Accessed 07 December 2014]. |
| [20] | Adafruit, "Thermocouple Type-K Glass Braid Insulated," Adafruit, [Online]. Available: http://www.adafruit.com/products/270. [Accessed 08 December 2014]. |
| [21] | Adafruit, "Adafruit 1-Wire Thermocouple Amplifier - MAX31850K," Adafruit, [Online]. Available: https://learn.adafruit.com/adafruit-1-wire-thermocouple-amplifier-max31850k/overview. [Accessed 08 December 2014]. |
| [22] | Adafruit, "4-channel I2C-safe Bi-directional Logic Level Converter - BSS138," Adafruit, [Online]. Available: http://www.adafruit.com/products/757. [Accessed 08 December 2014]. |
| [23] | Maxim Integrated, "DS18B20 Programmable Resolution 1-Wire Digital thermometer Datasheet," Maxim Integrated, [Online]. Available: http://datasheets.maximintegrated.com/en/ds/DS18S20.pdf. [Accessed 12 December 2014]. |
| [24] | Melexis, "MLX90614 family Single and Dual Zone Infra Red thermometer in TO-39 datasheet," Melexis, [Online]. Available: http://www.adafruit.com/datasheets/MLX90614.pdf. [Accessed 08 december 2014]. |
| [25] | Arduino, "Arduino playground: Reading a thermistor," Arduino, [Online]. Available: http://playground.arduino.cc/ComponentLib/Thermistor2. [Accessed 11 December 2014]. |
| [26] | Adafruit, "Using Melexis MLX90614 Non-Contact Sensors," Adafruit, [Online]. Available: https://learn.adafruit.com/using-melexis-mlx90614-non-contact-sensors. [Accessed 12 December 2014]. |
| [27] | Parr Instruments, "Plain Jacket Oxygen Bomb Calorimeter Instruction Manual," [Online]. Available: http://aui.ma/personal/~S.ElHajjaji/Labmanual/MSDS/InstructionManuals/BombCalorimeterManual2.pdf. [Accessed 14 12 2014]. |
| [28] | haake, "Haake Circulators Instruction Manual," [Online]. Available: http://www.geminibv.nl/labware/haake-q-koelwaterbad/haake-f3-series-circulators-manual-eng.pdf. |
| [29] | ReRap, "PCB Heatbed," [Online]. Available: http://reprap.org/wiki/PCB\_Heatbed. [Accessed 30 12 2014]. |
| [30] | J. Tyndall, Heat Considered as a Mode of Motion, London: Longmans, green and Co., 1868. |
| [31] | Kitt Peak National Observatory, "Building a Thermoelectric Sensor to Measure IR Sky Transparency for Cloud, Cirrus and Dust Detection, at Night," [Online]. Available: http://www.noao.edu/staff/gillespie/projects/cloud-detector.html. [Accessed 02 01 2015]. |
| [32] | Adafruit, "Adafruit ultimate GPS logger shield," [Online]. Available: https://learn.adafruit.com/adafruit-ultimate-gps-logger-shield. [Accessed 02 01 2015]. |
| [33] | Science Miseim UK, "Warming world: Satellites and weather balloons," [Online]. Available: http://www.sciencemuseum.org.uk/climatechanging/climatescienceinfozone/exploringwhatmighthappen/2point1/2point1point2.aspx. [Accessed 02 01 2015]. |