# DATA LOGGING A WATER ROCKET WITH ARDUINO


**Tim Joris [1], Jo-Hendrik Thysen [1], Michel De Roeve [1], Dirk Geeroms [1], Sabine Bertho [1]**

*[1] Hasselt University - Campus Diepenbeek, Agoralaan – Building D, 3590 Diepenbeek, Belgium*
*Email: michel.deroeve@uhasselt.be*



## ABSTRACT

Water rockets can achieve great acceleration and height. The takeoff of these rockets, i.e. the period during which the rocket is accelerated upwards, typically lasts 200 ms. To accurately visualize e.g. the acceleration during takeoff, a measuring frequency of 1000 Hz is desired. In this project, we show that this is possible using a fairly simple, student friendly micro-controller, namely the Arduino nano 3.0.


## 1. INTRODUCTION

When making a water rocket, a lot of effort is put in the design in order to reach a great height. The world record height is 609 m and during this flight a speed of 200 km/h was achieved [1]. When launched, the acceleration of a water rocket can exceed 100 *g*. The aim of this project is not reaching the greatest height with a water rocket but rather to adequately record the height and acceleration of the water rocket during its flight.

## 2. EXPERIMENTAL SETUP

The starting point for assembling the water rocket with a measuring unit is Arduino [2], an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. An Arduino board can be combined with several components that are capable of measuring the desired quantities.

### 2.1. Components

For the measurement of the acceleration, the ADXL001-70 (Analog Devices) was chosen. This accelerometer has a range of -70 *g* to +70 *g* and has an accuracy of 1 *g*. It is a capacitive acceleration sensor. Its working principle is depicted in Fig. 1. A movable frame is attached with 2 springs. When accelerated, the frame will move a distance *x* with respect to the capacitors according to Hooke's law:

$$F = k \cdot x \tag{1}$$

where *k* is the spring constant and *F* if the spring force. The 2 capacitors in the accelerometer (Fig. 1) measure the distance *x* that the frame has moved by the change in their capacitance. The acceleration *a* is then deduced from Newton's second law:

$$a = k \cdot x / m \tag{2}$$

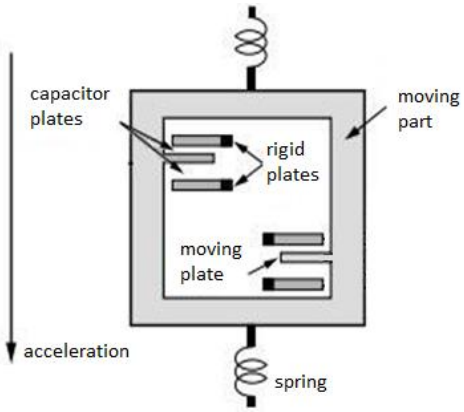


*Figure 1. A capacitive accelerometer.*

For the measurement of the altitude, the altimeter module MS5607 from Parallax was used. This device uses measurement of the air pressure *P* and temperature *T* to determine the altitude *h* with an accuracy of 20 cm:

$$h = \frac{T_0}{L} \cdot \left(\frac{P}{P_0}\right)^{\frac{-L \cdot R}{M \cdot g}} - \frac{T_0}{L} + h_0 \tag{3}$$

where $T_0$ is the temperature at sea level, *L* is the change in temperature per length unit in the troposphere, $P_0$ is the pressure at sea level, *R* is the universal gas constant, *M* is the molar mass of air, *g* is the gravitational constant and $h_0$ is the altitude at sea level.

To make sure that the rocket and the accompanying measuring equipment land safely, a parachute mechanism was designed. This mechanism consisted of a compass module, a cap containing a parachute and a servo motor to release the cap. The compass module CMPS10 from Robot Electronics is capable of measuring the orientation of the rocket using the magnetic field of the Earth. The microcontroller was programmed to trigger the servo motor when the rocket tilts more than 45°. This is shown in Fig. 2.

The Tower Pro Micro 9 gram SG90 servo motor can rotate over a maximum of 180°. When switched on, it will release a parachute by opening a cap. Inside the


*Proc. '21st ESA Symposium European Rocket & Balloon Programmes and Related Research',*
*9–13 June 2013, Thun, Switzerland (ESA SP-721, October 2013)*

cap, a small parachute is attached to a large parachute. The small parachute will first leave the cap and pull the large parachute out of the cap.

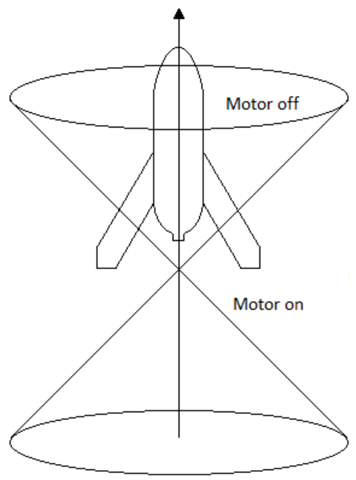


*Figure 2. Once the rocket points out of the cone, the servomotor is switched on and the parachute is released.*

### 2.2. Assembly of the measuring unit

The above-mentioned components were connected to 2 Arduino Nano 3.0 boards. One board was connected to the accelerometer, the altimeter and an microSD board (Fig. 3). A 16 GB microSD-card was inserted in the microSD board for the data storage. The second Arduino Nano 3.0 was connected to the compass and the servo motor as shown in Fig. 4.

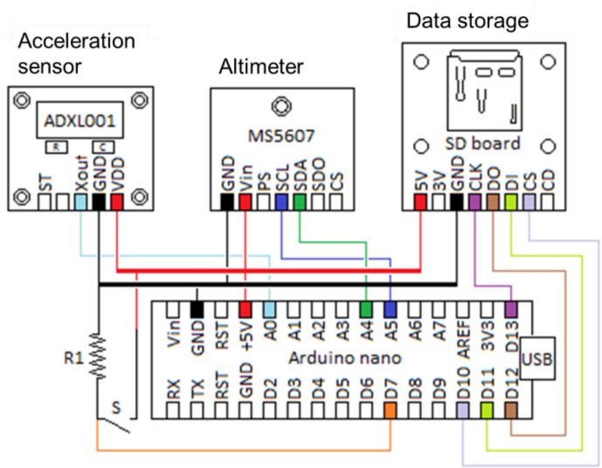


*Figure 3. An Arduino Nano 3.0 connected to the accelerometer, the altimeter and an microSD-board. R1 is a resistance of 10 kΩ and S is a switch.*

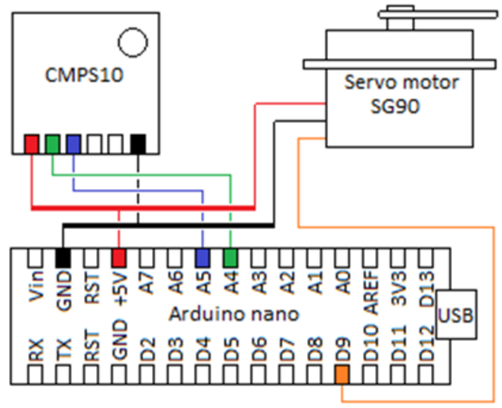


*Figure 4. An Arduino Nano 3.0 connected to the compass and the servo motor.*

The whole measuring unit was united in a circular cap that was mounted on top of a water rocket. The measuring components were distributed across 3 PCBs (printed circuit boards). The bottom PCB (Fig. 5) contained two 9 V batteries (each with its own switch) for powering the Arduino Nano 3.0 boards. The middle PCB (Fig. 6) contained a switch for starting the measurement, an Arduino Nano, the microSD board and the altimeter. The top PCB (Fig. 7) contained the second Arduino Nano, the compass, the accelerometer and the servo motor.

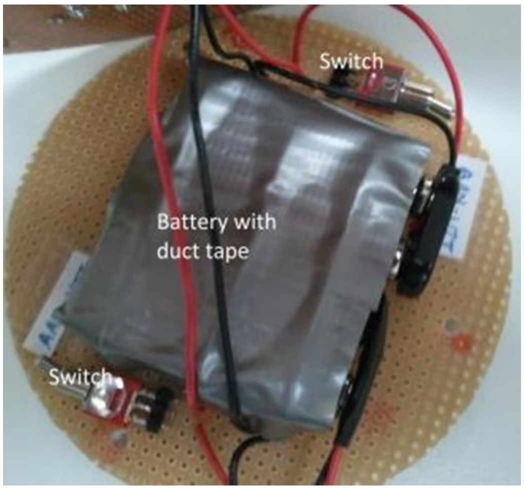


*Figure 5. The bottom PCB contained two 9V battery, each with its own switch.*

When mounted on top of each other, the complete measuring setup looked like Fig. 8. This measuring unit was attached on top of the water rocket as in Fig. 9. A cap that could be opened by the servo motor contained the parachute. A smaller parachute was intended to pull out the larger parachute.

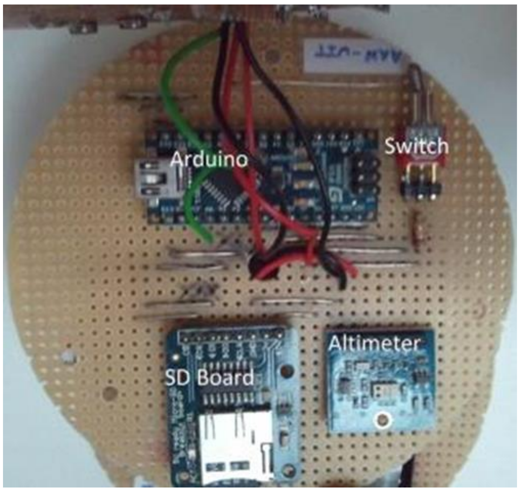


*Figure 6. The middle PCB contained a switch, an Arduino Nano, the microSD board and the altimeter.*

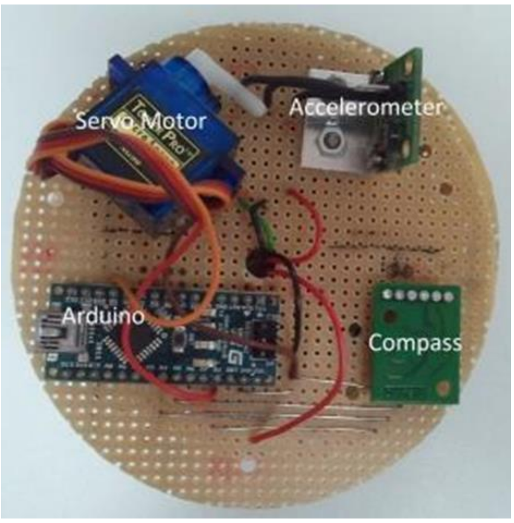


*Figure 7. The top PCB contained an Arduino Nano, the compass, the accelerometer and the servo motor.*

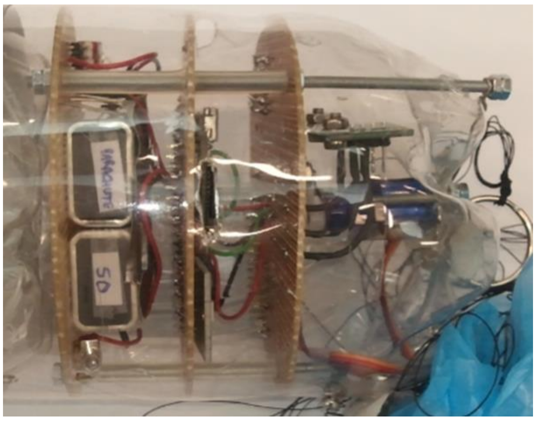

*Figure 8. The whole measuring unit put together in a circular cap.*

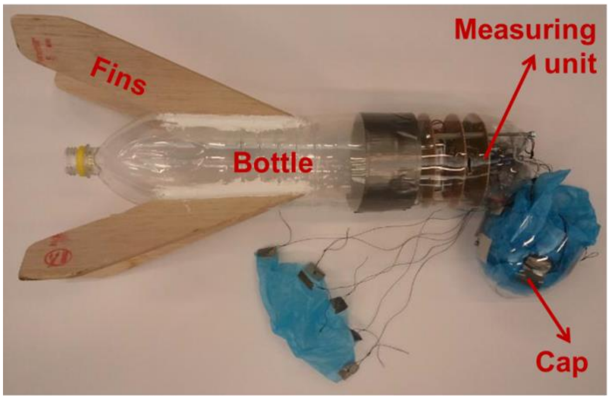


*Figure 9. The water rocket with the measuring unit attached on top. The parachute was inserted in a cap that could be opened by the servo motor.*

### 2.3. Software

The open-source Arduino software was used to program the measuring setup. The C-based Programming Language is called Processing. The code had 2 main parts. It started with a *void setup* for initializing all variables. The actual measurements were programmed in the *void loop*. The code could be uploaded to the Arduino via USB communication.

An important requirement for the software was the high measuring frequency. The water driven phase of a water rocket typically lasts only 200 ms, making a measuring frequency of about 1000 Hz desirable for accurately monitoring the acceleration during this phase. The altimeter performed quite complex calculations (Eq. 3), therefore each measurement of the altimeter lasted 89 ms and with this it was not possible to achieve the desired measuring frequency of 1000 Hz. Therefore, the altimeter was programmed to start measuring after the initial launch phase. In this way it was possible to measure the acceleration during takeoff with a high measuring frequency. Once the propulsion due to the ejecting water had ended, the rocket only accelerated due to gravity and the fast measurement of the acceleration was no longer necessary.

Another crucial point that took time during the measurement was the writing command from the Arduino to the SD card. In a standard code, each measurement point was written to the memory card separately. A lot of time could be saved by storing data in the Arduino board until its memory was full and then flushing this memory as a large data block to the memory card. With this adjustment in the code, it was possible to record one measurement every 0,8 ms. This means that a measuring frequency of 1250 Hz was achieved which was high enough to monitor the acceleration of the water rocket during take-off.

## 3. RESULTS

### 3.1. Height measurement

The height sensor was tested with a very simple experiment. The height of a room was measured either with a tape-measure or by holding the height sensor at the floor and the ceiling of the room. The tape-measure indicated that the room had a height of (2.41 ± 0.05) m and the height sensor gave a result of (2.5 ± 0.4) m. Both results corresponded so the height sensor was properly installed and correct results could be expected.

### 3.2 Acceleration measurement

The acceleration sensor was tested by dropping the rocket from the third floor of a building. The results are shown in Fig. 10. As can be seen in the graph, there were a lot of fluctuations on the acceleration but the average values of the acceleration (full lines in the graph) correspond to plausible values of the acceleration.

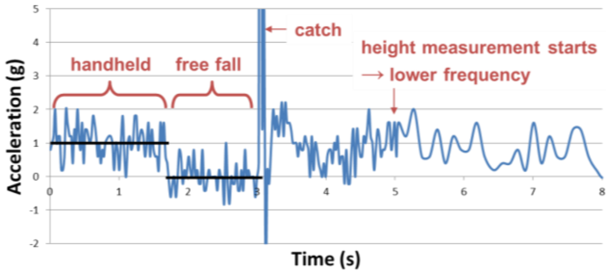


*Figure 10. Free fall of the acceleration sensor from the third floor.*

At first sight this did not seem to be the case. When the rocket was handheld an acceleration of 0 *g* was expected but an acceleration of about 1 *g* was measured. When the rocket was in free fall the acceleration sensors measured 0 *g* while 1 *g* was expected. The layout of the acceleration sensor (Fig. 1) gave the explanation for this phenomenon. The displacement of a movable frame with respect to a fixed frame is a measure for the acceleration. When the sensor was handheld, the movable frame moved a little out of its equilibrium position until the spring force balanced gravity. This small deviation thus corresponded to an acceleration of 1 *g*. When the rocket was in free fall, both the movable and the fixed frame experience the same acceleration due to gravity and the movable frame stayed in its equilibrium position with respect to the fixed frame. The acceleration sensor thus registered an acceleration of 0 *g*.

Furthermore, Fig. 10 shows a sharp acceleration peak around 3 s. At this point the rocket was caught, which corresponded to a very high acceleration: a value of 47.4*g* was registered (not shown in the graph).

In paragraph 2.3. it was said that the measurement of height started after a certain time delay to increase the measuring frequency during takeoff of the rocket. This is visible in Fig. 10 where 2 seconds after the acceleration peak, which acted as a trigger point for the height measurement, the measuring frequency clearly slowed down.

### 3.2. Rocket launch

The water rocket was launched with a homebuilt system. The rocket was mounted on a launch tube where it was attached firmly to prevent it from launching too early. With a simple air pump, the pressure inside the water rocket was increased. For the launch of the rocket, an inside pressure of 8 bar was chosen before the rocket was allowed to take off. During the first actual flight of the water rocket, the parachute mechanism opened at the right moment but unfortunately, the parachute itself got stuck behind a screw inside the parachute cap. This resulted in a fatal crash of the water rocket and the measuring equipment. However, the microSD card could still be retrieved from the measuring unit.

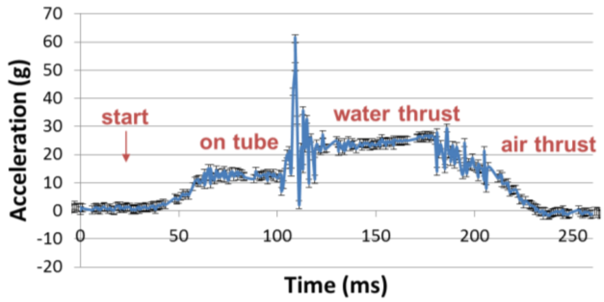


*Figure 11. The measured acceleration during takeoff of the water rocket. Three phases can clearly be distinguished.*

The acceleration during takeoff (Fig. 11) was correctly stored on the memory card but no data concerning the height could be retrieved. A possible explanation for this error could be the data transmitting process where data was not flushed to the memory card until a data block was full. If the data block for the height measurement was not full yet at the moment of the crash, it would not have been transmitted to the memory card.

Because of the high measuring frequency of the software, it was possible to very accurately record the acceleration during takeoff. In Fig. 11 three phases can clearly be distinguished. During the first phase the water rocket moved along the launch tube. The tube limited water flow, which kept the acceleration on average at $(11.4 \pm 0.4)$ $g$. When the rocket left the tube, the acceleration peaked at $(60 \pm 1)$ $g$. During the second phase the rocket lost all of its water. On average the acceleration was $(23.0 \pm 0.3)$ $g$. Due the decreasing mass of the rocket, the acceleration increased slightly. In the third phase, high pressured air left the rocket and caused thrust. As the pressure decreased, so did the acceleration. The three phases combined took about 190 ms.

## 4. CONCLUSION

This manuscript described a project carried out by 2 students during a time period of 15 days spread across 10 weeks. It shows that in this short time period it was possible to achieve nice results concerning the launch of a water rocket. The open-source platform Arduino was the base of this project and proved to be very student friendly. It was possible to build a measuring setup that can record the height and acceleration of a water rocket during its flight. Unfortunately, the parachute got stuck during the first launch (though the parachute mechanism had worked correctly) but still nice results concerning the acceleration during takeoff could be recorded. Due to a well written program with a high measuring frequency of 1250 Hz, it was possible to clearly visualize the three phases of the rocket launch.

## 5. REFERENCES


1. Official water rocket world record standings, URL: http://www.wra2.org/WRA2_Standings.php

2. Arduino Blog - The Arduino Documentary, URL: http://arduino.cc/blog/2012/02/20/the-arduino-documentary-2/