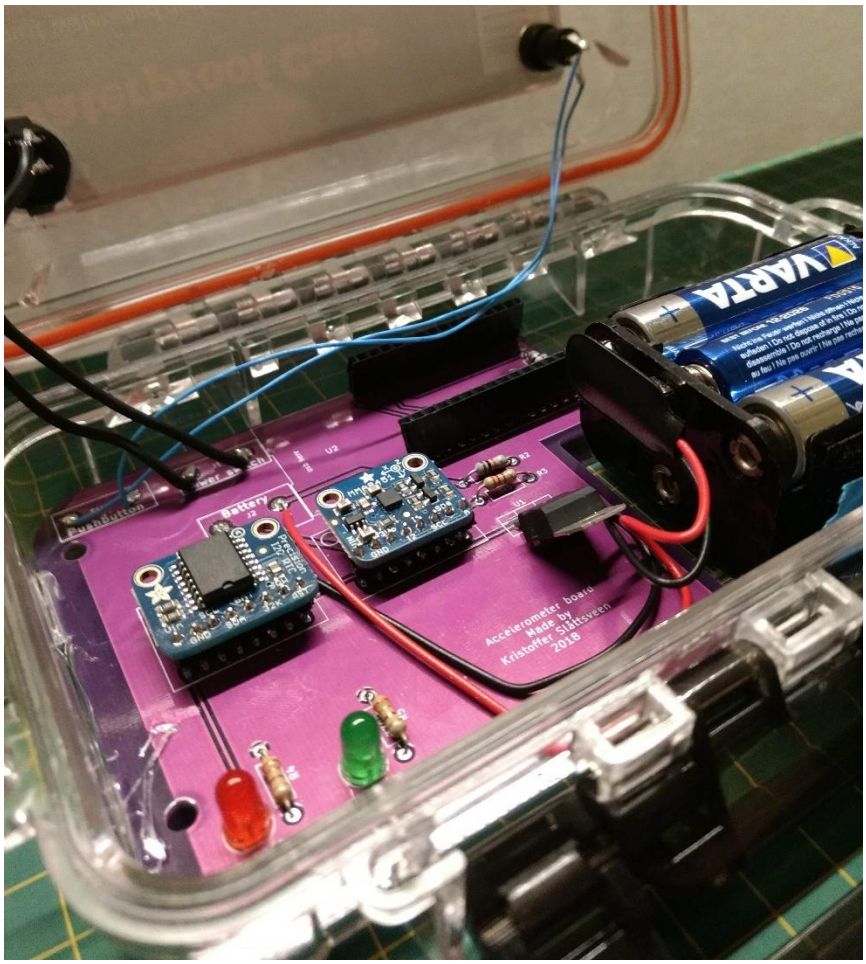
Arduino™ Accelerometer Logger

How to measure dynamics of structures



Written by:

Kristoffer Slåttsveen

2018

(empty page)

Arduino™ Accelerometer Logger – How to!

Preface

This project has been taken on with the goal of creating a DIY accelerometer-based datalogger for civil engineering students. The projected use case is measuring dynamics of constructions such as bridges and large buildings.

Table of Contents

Glossary

Soldering – The process of connecting electrical components by melting and applying solder to a joint.

Solder – A metal alloy that melts at relatively low temperatures (280-250 degrees C) used for soldering electrical components together.

Arduino – An easy to use family of microcontrollers, empowering the average Joe to creating electronics projects.

IDE – Integrated Developing Environment, basically a program to write code. The Arduino IDE is made specifically for the Arduino family of boards. Arduino is a registered trademark!

PCB – Printed Circuit Board, it is the plate that connects all our components together and holds them in place

Sketch – We call a program written for the Arduino a Sketch. The Arduino can only hold one program at a time.

1 – Preparation

In order to be prepared for this project, the following prerequisites need to be met: 1) Read this tutorial. 2) Purchase all the parts. 3) Have access to soldering equipment and some way of producing holes in a plastic box, such as a drill with the required bits. 4) If you have never done any soldering before, review a tutorial video or have someone show you how it is done. 5) Install the Arduino programming IDE on your computer.

Hopefully someone has already prepared the materialistic part of the preparations for you, further details will however be presented further along in this document.

After acquiring all the necessities, read through the instructions and get going! The project should take ca 1.5 hours to complete. 6) Build the datalogger. 7) Download the sketch written for this project and program the Arduino. 8) Gather all the data!

2 – Component list

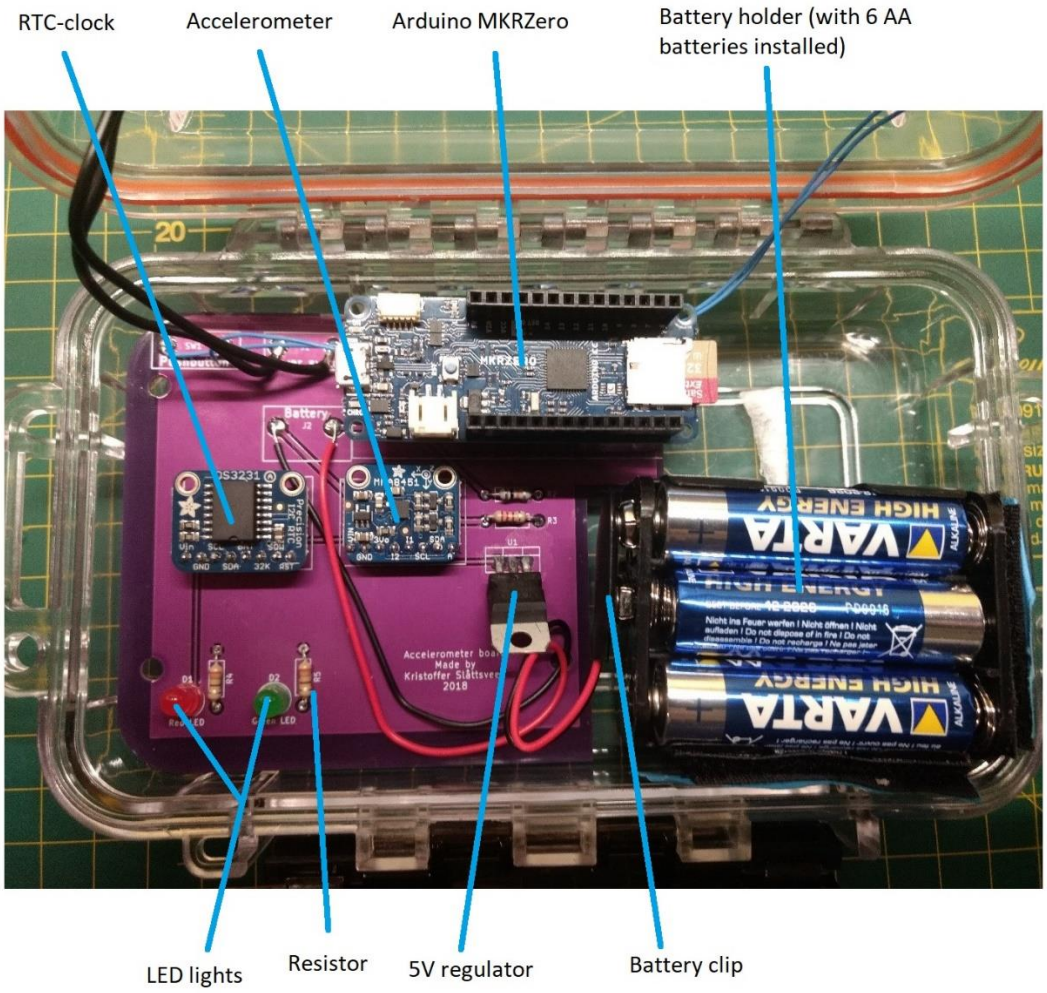
These are the parts needed to build the datalogger described in this document. At the end of this chapter, you find a picture with the names of the parts explained.

Table 1: BOM (Bill Of Materials)

Component name	**Number of items**	**Price [NOK]**	**Link to product** Any retailer selling the same product is fine, this is only an option!
Arduino MKRZero The brain of the project	1	243	https://www.elfadistrelec.no/en/arduino-mkr-zero-arduino-abx00012/p/30101961?queryFromSuggest=true
Accelerometer MMA8451 Our main sensor	1	91	https://www.elfadistrelec.no/en/mma8451-accelerometer-board-adafruit-2019-accelerometer/p/30091168?queryFromSuggest=true
RTC DS3231 Real Time Clock, it keeps track of time when the logger is powered down.	1	156	https://www.elfadistrelec.no/en/ds3231-precision-rtc-adafruit-3013-ds3231-breakout/p/30091211?queryFromSuggest=true
Toggle switch Main power switch	1	13,6	https://www.elfadistrelec.no/en/ultraminiature-power-rocker-switch-black-on-off-rnd-components-rnd-210-00543/p/30109750?q=*&sort=Price:asc&filter_Switching+type%2C+%28+%29momentary=On-Off&filter_Termination+type=Soldering+Connection&filter_Buyable=1&filter_Category2=Switches&page=4&origPos=638&origPageSize=25&simi=100.0
Pushbutton This will trigger the datalogging	1	14,6	https://www.elfadistrelec.no/en/miniature-pushbutton-switch-monacor-312-rt/p/11034095?q=*&filter_Poles=1&filter_Switching+type%2C+%28+%29momentary=Off-%28On%29&filter_Buyable=1&filter_Category2=Switches&page=3&origPos=845&origPageSize=25&simi=99.9
Red LED Power indicator light	1	1	https://www.elfadistrelec.no/en/led-mm-round-red-rnd-components-rnd-135-00032/p/30100029?q=*&filter_Category2=LEDs&filter_Light+colour=Red&filter_Case+colour=Red&filter_categoryCodePathROOT%2Fcat-L2-3D_525341%2Fcat-L3D_525297=cat-DNAV_PL_020103&filter_Buyable=1&filter_categoryCodePathROOT%2Fcat-L2-3D_525341=cat-L3D_525297&filter_Housing+type=Diffuse&filter_categoryCo

Component name	**Number of items**	**Price [NOK]**	**Link to product** Any retailer selling the same product is fine, this is only an option!
Green LED Indicator light for datalogging	1	1	https://www.elfadistrelec.no/en/led-mm-round-yellow-green-rnd-components-rnd-135-00027/p/30100024?q=*&filter_Category2=LEDs&filter_Case+colour=Green&filter_categoryCodePathROOT%2Fcat-L2-3D_525341%2Fcat-L3D_525297=cat-DNAV_PL_020103&filter_Buyable=1&filter_Lens+size=5+mm+%28T13%2F4%29&filter_Lens+size=5+mm+round&filter_categoryCodePathROOT%2Fcat-L2-3D_525341=cat-
Resistor 330Ohm	2	0,7	https://www.elfadistrelec.no/en/carbon-film-resistors-330-ohm-25-yageo-cfr-25jt-52-330r/p/16050226?q=*&sort=Price:asc&filter_Category3=Through+Hole+Resistors&filter_Buyable=1&filter_Resistance~~Ohm=330&page=2&origPos=720&origPageSize=25&simi=99.9
Resistor 10kOhm	2	0,7	https://www.elfadistrelec.no/en/carbon-film-resistors-10-kohm-25-yageo-cfr-25jt-52-10k/p/16050244?q=*&filter_Category3=Through+Hole+Resistors&filter_Resistance~~kOhm=10&filter_Buyable=1&page=2&origPos=149&origPageSize=25&simi=99.9
Resistor 2k4 Ohm	1	1	https://www.elfadistrelec.no/en/resistor-kohm-vishay-mba02040c2401fct00/p/16059399?q=*&sort=Price:asc&filter_Category3=Through+Hole+Resistors&filter_Resistance~~kOhm=2.4&filter_InStock=2&filter_InStock=1&filter_Buyable=1&page=2&origPos=1953&origPageSize=25&simi=99.9
Female header Connection point for the Arduino	2	10	https://www.elfadistrelec.no/en/single-row-straight-without-shroud-female-header-female-14-wuerth-elektronik-61301411821/p/30024905?q=female+header&sort=Price:asc&filter_Poles=14&filter_Connector+Design=Single+Row&filter_Connector+Design=Straight&filter_Pitch~~mm=2.54&filter_Connector+Termination=THT&filter_Lines=1&page=1&origPos=305&origPageSize=25&simi=98.96
USB cable For programming the Arduino and streaming data	1	32	https://www.elfadistrelec.no/en/usb-cable-black-valueline-vlcp60500b10/p/30046369?q=*&sort=Price:asc&filter_Cable+type=USB+2.0&filter_Connection+A=USB-A+Male&filter_Connection+B=USB-Micro-B+Male&filter_Buyable=1&filter_Category2=Cable+Assemblies&page=1&origPos=202&origPageSize=25&simi=99.9
Battery clips	1	7	https://www.elfadistrelec.no/en/battery-holder-9v-keystone-233/p/16914329?q=battery+holder&sort=Price:asc&filter_Suitable+for=9V&page=3&origPos=141&origPageSize=25&simi=98.88
5V regulator Brings the battery voltage down to 5V for the Arduino	1	7	https://www.elfadistrelec.no/en/ldo-voltage-regulator-to-220-lm2931-texas-instruments-lm2931t-nopb/p/17327435?q=*&filter_Category2=ICs&filter_Output+voltage=5+V&filter_Buyable=1&filter_Package=TO-220-3&page=1&origPos=968&origPageSize=25&simi=99.9
6xAA battery holder	1	15	https://www.elfadistrelec.no/en/battery-holder-aa-comf-bh343-1b/p/16955330?q=*&filter_Termination+Type=Prong+Snap&filter_Battery+Holder+Type=Holder&filter_Suitable+for=AA&filter_Buyable=1&filter_Category2=Batteries+and+Chargers&page=3&origPos=702&origPageSize=25&simi=99.9
Velcro strap Self adhesive, for fastening the battery holder. (ca 6cm is required pr. logger)	0,02	120	https://www.clasohlson.com/no/Borrel%C3%A5sb%C3%A5nd/Pr344680001
Waterproof box Something to keep it all in (we'll be drilling holes in it)	1	80	https://www.clasohlson.com/no/Vanntett-boks/31-8543

Component name	Number of items	Price [NOK]	Link to product Any retailer selling the same product is fine, this is only an option!
SD card For storing data, needs to be class 10 at least	1	149	https://www.japanphoto.no/sandisk-microsdhc-extreme-pro-16gb-95mb-s
PCB The board that makes everything look professional! Made on demand.	1	34	www.pcbway.com



3 – Tools

These are the tools you need to make the datalogger.

Table 2: Required tools and equipment

Soldering iron – almost anything will do here, 25W and up.
Solder
Flux (or flux core solder)
Brass sponge or damp sponge – for cleaning the soldering iron
Wire cutters (flush cutters are preferred)
Wire (18-20 AWG for example)
Helping hands – Nice to have
Desoldering wick or solder sucker – Nice to have
Drill
7mm drill bit
15mm drill bit – or a step drill bit (the one that looks like a Christmas tree)
Hot glue gun
Computer with 1 USB port and an SD-card reader

4 – Learning to solder

If you know how to, skip and dance past this section, or teach someone sitting next to you how to get it done! For the rest of you, there is always YouTube: A suggested soldering tutorial is provided in various formats here:

A very thorough but quite short video:
https://www.youtube.com/watch?v=IpkkfK937mU

If you are reading this how-to printed out, the link may be hard to click. As an alternative solution, scan this QR-code with your smartphone and learn all about soldering through a tiny screen!

5 – Installing the Arduino IDE

This process is quite simple, you go to a website and download the appropriate installer for your computer. Mac, Windows or Linux. In addition to installing the IDE, we will be using a slightly more specialized Arduino board than what is typically used for small learning projects. Therefore we also need to install the correct drivers for this board (the Arduino MKRZero).

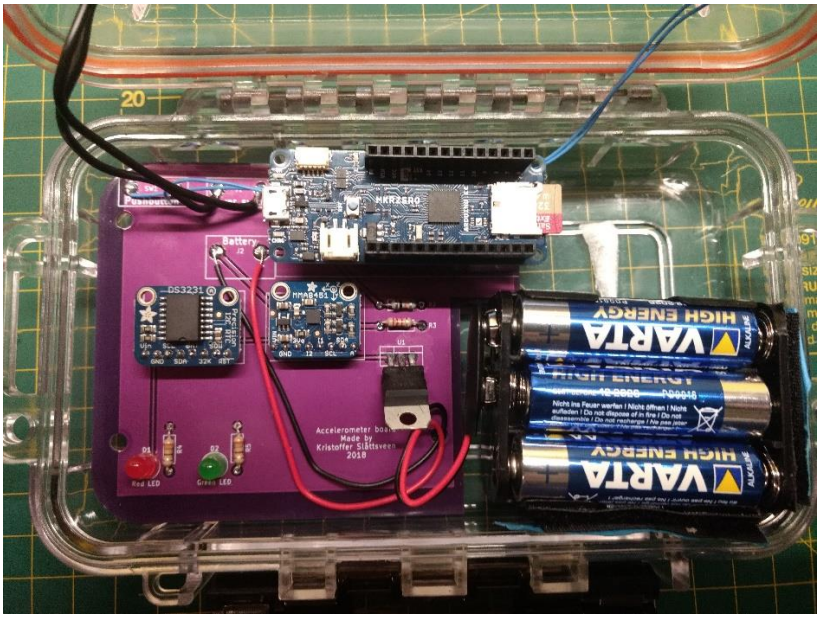
I'll take you through the steps:

1) Go to www.arduino.cc – then click "Software" and "Downloads" to find the download link for your computer.
2) Once the program is installed, open it and connect your Arduino MKRZero via the Micro USB cable.
3) Install drivers, we have two options here.
    a. Automatic detection - After a little while, a small window should pop up down on the left side of the IDE suggesting you install support for some new boards. Click and follow instructions.
    b. Manual installation – Go to: Tools (Verktøy) -> Board: (Kort:) -> Boards manager… (Kort administrasjon…). A window with a search bar should open, search for "MKRZero" and install the only option left.

6 - Build the datalogger

Building the datalogger consists mainly of assembling the electronics, as well as drilling some holes for the buttons in the casing. The end product should look something like what you see in these two pictures:

The two buttons can be placed more or less everywhere on the casing. The only requirement is that they don't interfere with the components on the inside of the box. And the wiring to the buttons should be long enough for the lid to be opened fully.



This picture shows the first version of the PCB (purple color). It has a few design errors, but we can work around those.
The 5V regulator must be mounted backwards, as shown in the picture.
The 3k6 Ohm resistor is replaced with a 2k4 Ohm resistor.
The pins on the Arduino and the headers on the board must be bent slightly for the Arduino to fit.

Before you start soldering, there are a few tips that is smart to think about. Don't place all the components on the PCB at the same time before you turn it over and start soldering. Do maximum a few parts at the same time. And it is a good idea to start with the smallest components first, i.e. the ones sticking the least amount up from the board. That way, when you flip it over, the parts are pressed between the table and the PCB, making sure it stays nice and low.

For components with more than two pins, it is a good idea to solder one pin first, then check alignment before you solder the rest of the pins. **Try to get the accelerometer to stay as parallel to the PCB as possible!** This to ensure that the axes of the accelerometer are closely aligned with the axes of the box.

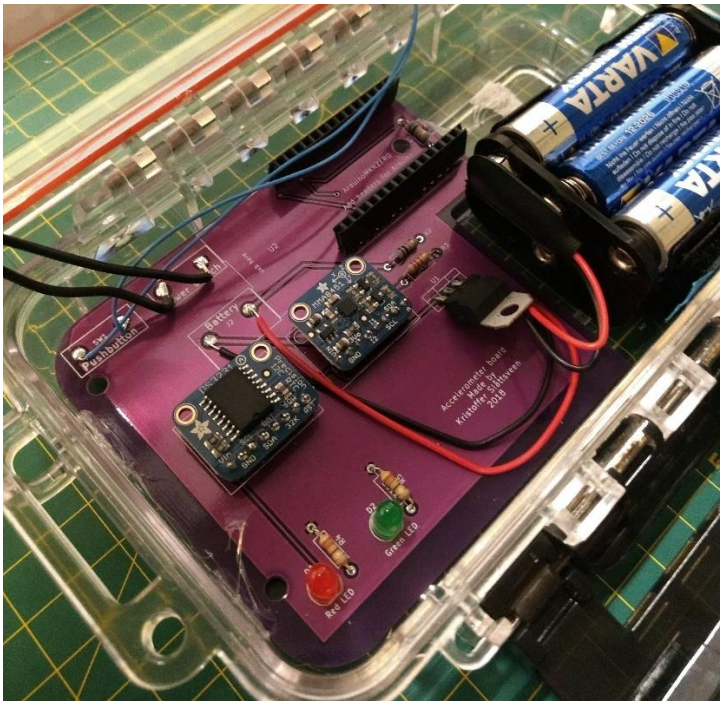
Solder on the wires to the buttons and the battery clip last. So you can measure up and cut the wires to a proper length for neat and tidy wiring. Pay attention to the polarity of the battery leads! Red goes to +, and black goes to – on the board. The best way to solder these wires to the board is to first apply solder to the wire ends and the pads separately, then laying the wire on the tinned pad, you solder them together.

Now that all the components are on the board, test-fit everything and decide on placement of the buttons. Then drill the appropriate holes and mount the switch and the button. The power switch is pressed into its hole from the outside, and the pushbutton is fastened with a nut after being placed through the hole from the inside.

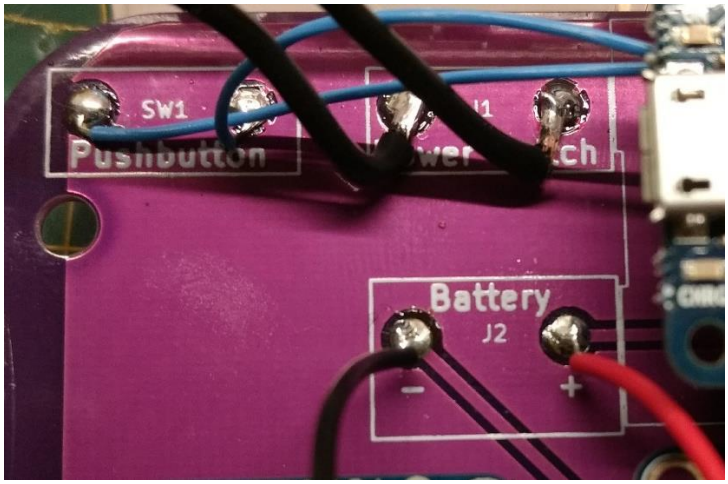
When the PCB is ready, the switches has been mounted and we have tested the datalogger, it is time to fasten things inside the box. Use hot-glue to fasten the PCB to the bottom of the box, make sure it is moved all the way to the side-wall. Lay down ca 2cm strips along the wall and the PCB.

Use the Velcro strap to fasten the battery to two walls of the box. It should be apparent which two walls to use. See the pictures below for reference. The Velcro straps that are put in the box is going to be wider than the height of the box wall. This can be accommodated by either placing some paper on the remaining area containing glue, or simply cutting it down with some scissors.

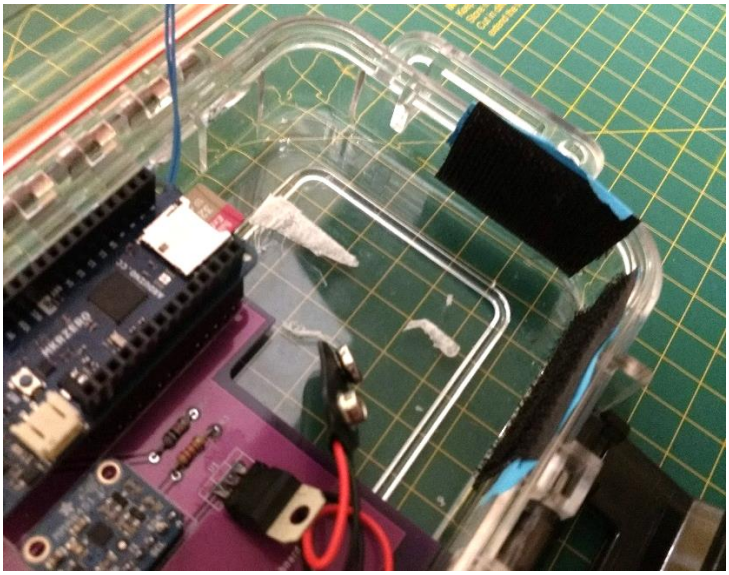
Some more pictures of the details:



Notice that there are headers that the Arduino slots into. This is so the Arduino is removable, and it raises it high enough for the USB-cable to get clearance of the box edge.



When soldering the wires to the pads for the button, power switch and battery clip. Only the battery clip is sensitive to polarity.



Place Velcro pieces on the sides of these two walls. If you find a good way of securing the battery even better, while still being able to remove it easily, feel free to implement it!

Place the matching Velcro pieces on the battery holder.

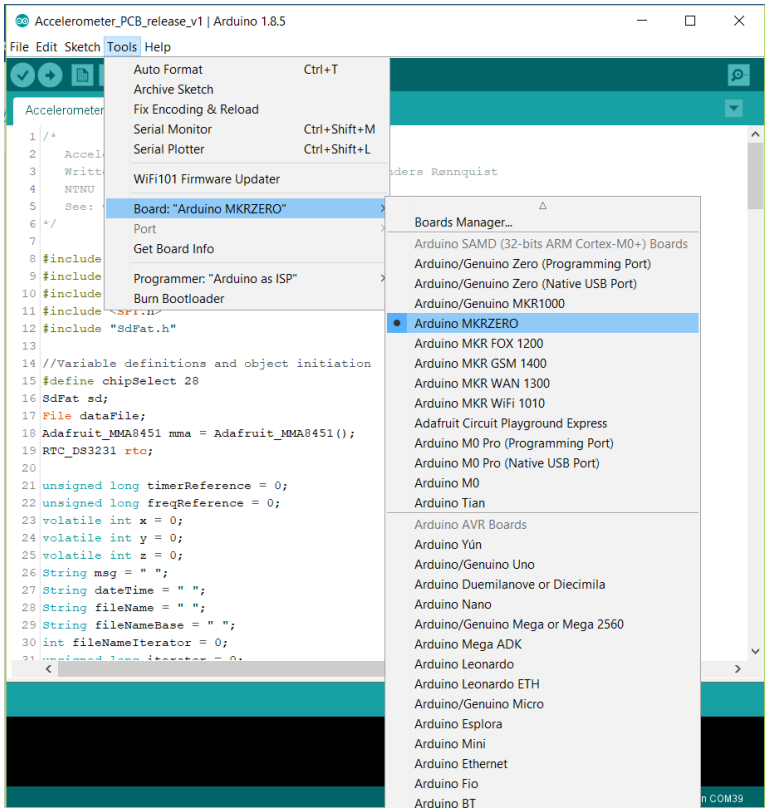
7 – Program the Arduino

The code that makes the Arduino put acceleration values on the SD-card for us has already been written. You find that piece of code on GitHub:
https://github.com/Slattsveen/Arduino_acceleration_logger

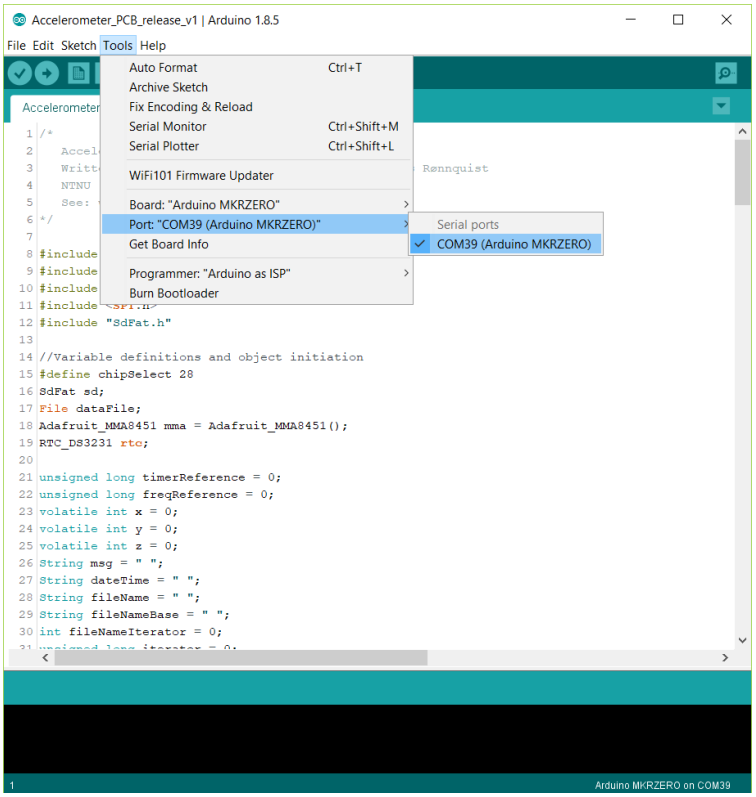
There is a .zip file there called Code.zip. It contains two programs, as well as some library files for the Arduino to use. These libraries allow us to read values from the accelerometer and talk to the RTC-board. One of the two programs are for the datalogging, the other one is just made for setting the time on the RTC-board. Start with the one for setting the clock, then we can overwrite that one with the datalogger code afterwards. The clock will remain set for as long as its battery lasts, ca 15 years or so ☺!

NB: The libraries must be moved from wherever you unzipped the Code-folder to your Arduino libraries folder. For a normal installation, that folder is found in "C:\Users\'user'\Documents\Arduino\libraries". Simply move the contents of the code\libraries (there is three folders there), to your ~Arduino\libraries folder. Then restart the Arduino IDE.

Programming the Arduino MKRZero is only a matter of opening the code-file you want to use, connecting the USB-cable, selecting the correct COM-port and Board from the Tools menu and clicking the upload button. See visual steps for navigating the Arduino IDE on the following pictures. If your computer has problems with recognizing the Arduino, it may be because the circuit wants to draw a little more current than the Arduino likes. (you see this by checking if the CHRG LED on the Arduino lights up, orange light next to the USB-port). In that case, just put in the batteries and turn on the power switch.



Select the Arduino MKRZero, this is to let the program know which board it is going to compile code for.

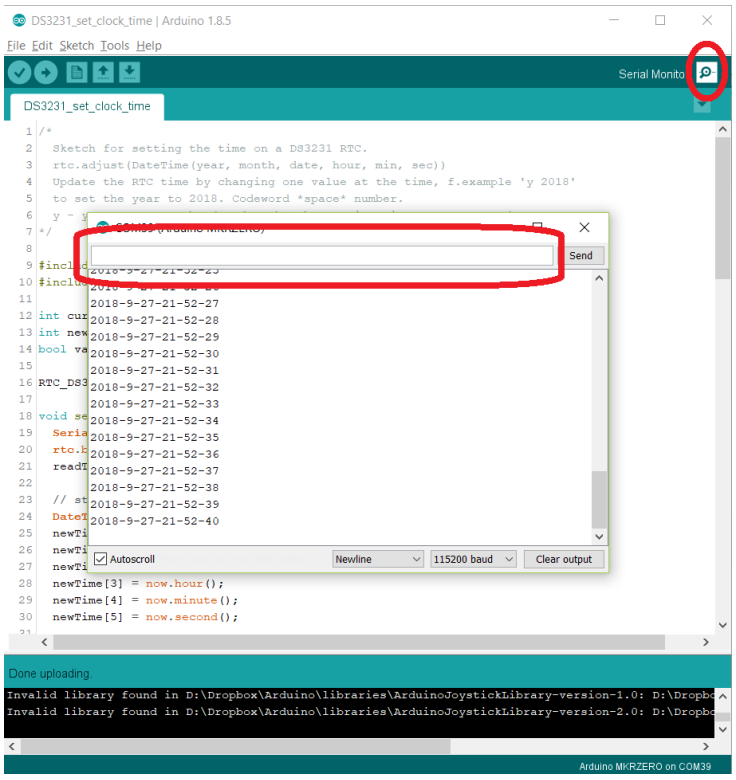


Select which ever port the Arduino is connected to, it should say so right next to the COM port. On Mac and Linux computers, the COM has another name, but it works exactly the same.

The Upload button is easy to find in the top left corner. Alternatively, you can be a super-user and press Ctrl+U as a keyboard shortcut.

Now that you have uploaded the “DS3231_set_clock_time” sketch, we can proceed with actually setting the clock! To do this, we connect the Arduino and open the Serial Monitor. The button is in the upper right side of the IDE and looks like a magnifying glass. From the Serial monitor we can see messages the Arduino sends to us, and we can send messages to the Arduino. Details on how you set the time is given in the code text. You can also type “H1” into the serial monitor to get some instructions.

Setting the Baud-rate; the Arduino code is set to communicate with the computer at a baud rate of 115200, make sure you set the same rate in the serial monitor. This option is found at the bottom right of the Serial Monitor window.



The Serial monitor is a separate window that shows messages from the Arduino, sort of like a terminal.

Once you have set the correct time and date, you can upload the “Accelerometer_PCB_release_v1” sketch and begin logging data!

8 - Gather all the data!

Now that the datalogger is finished it is about time to head out and gather some data!

Before that though, we may want to test it and see that everything is as it should. First off, install the SD-card and the batteries. Then to the controls; to power on the device, turn on the power switch. This should make the red LED turn on. After a short startup sequence (ca 3-4 seconds), we can start logging data by pressing the pushbutton once. This should turn on the green LED, indicating that a datalogging sequence is going on.

To turn it off, we reverse the sequence; press the pushbutton to stop the recording of data. **This is important!** If the recording is not properly shut down, the whole dataset will be lost. Always make sure the green light is OFF before killing the power. In case the batteries run low, the logger will stop collecting data, and the green light will start blinking.

To review the data, power down the Arduino and remove the SD-card. Install it in your computer and look up the relevant file. The filenames are generated automatically based on the time of the measurement, something like this: "27d9m21H22M", translating to the 27th day of the 9th month at 21 hours 22 minutes. -> 21:22 on the 27th of September.

The contents of the datafile will look like this:

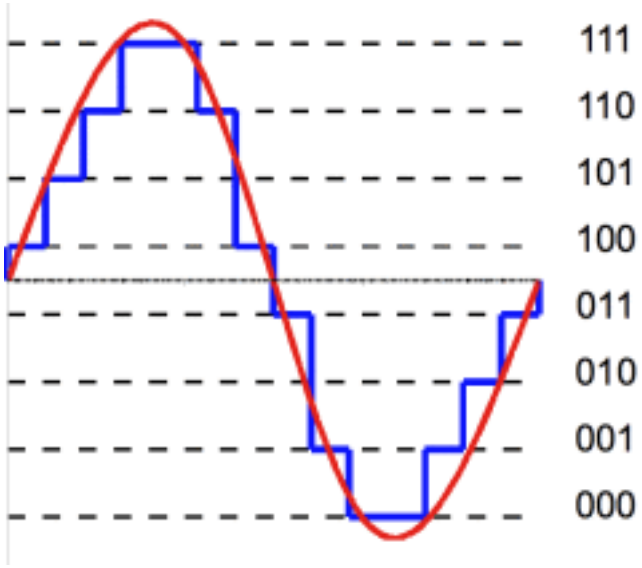
```
File opened: 27/9/2018 21:22:21
#, x, y, z, dt[microseconds]
0,-88,26,4052,1417
1,-108,10,4054,1998
2,-84,16,4028,1999
3,-106,8,4030,2003
4,-88,10,4058,1999
5,-96,12,4048,2000
6,-96,12,4060,1999
7,-84,12,4042,2000
8,-90,18,4046,2000
9,-92,26,4046,2000
10,-90,14,4060,2000
.
.
.
.
.
.
3430,-84,36,4032,2000
3431,-82,14,4052,2000
3432,-102,20,4048,2001
3433,-88,36,4078,1999
3434,-98,30,4056,1999
3435,-92,14,4038,3788
3436,-92,20,4066,1999
3437,-90,0,4058,2000
File closed: 27/9/2018 21:22:28
```

It can be imported as a .csv file into excel or other spreadsheet programs. There are 5 columns in the file, and the header explains what those columns contain. The first is a measurement number,

followed by acceleration values in x, y and z, direction, and lastly the time difference between the current measurement and the previous one (dt) in microseconds.

A word about the acceleration values. This is a 14-bit accelerometer with a +-2g sensitivity, which means that the g-value it measures is measured with a resolution of 14-bit. In practice, the sensitivity of +-2g (4g from end to end), is spread out over a scale with $2^{14} = 16384$ discrete steps. The way the Arduino (and other programs) store numbers, reduces this scale by 1, giving us a scale with a length of 16383 steps to measure the +-2g the sensor can sense. Quite precise in other words.

The number you see in the dataset, will be somewhere on that scale from -8192 (-2g) to +8191 (+2g).



This figure represents a 3-bit resolution laid over a continuous curve. Our discrete curve would look much smoother, since there would be 16383 steps from top to bottom of the sine-wave.

Bonus content!

If you don't have an SD-card, or simply want to read data straight into your computer. You can! You will however have to do some scripting on your own. The Arduino sends a Serial stream to the computer, which can be read live with various programming or scripting languages, Matlab and Python being the easiest ones.

Use the Arduino IDE to check which COM-port the Arduino is connected to, and set up a Serial read function in your scripting language of choice. From there on, it is up to you what you do with it!

The Arduino automatically jumps into this state if there is no SD-card installed when the Arduino is powered up. You do probably need to have the batteries installed and the power switch turned ON for the datalogger to work properly.

Good luck, and happy logging!