

ARDUINO WEATHER STATION

ANDREW LORIMER

PART A – CONSTRUCTION RECORD

PLANNING – WRITTEN FEB 2016

Originally I planned on using a pre-release version of the [CHIP microcomputer](#) (a US\$9 SBC, similar to an RPi) which I already own. However I have decided to use [Arduino Nano](#) microcomputers for both the tx and rx units, due to their community support, codebase and documentation.

I will probably follow most of this tutorial (~~using analog panel meters instead of a LED display~~) - <http://www.instructables.com/id/Arduino-weather-station/?ALLSTEPS>. I will also add wind speed/direction sensors if I have time. I could also allow the receiver unit to upload the data to a web server (possibly [IBM's IoT platform](#)) so that it is visible on a computer/phone (this is also dependent on time).

This tutorial will be useful if I get time to implement wifi - <https://learn.adafruit.com/wifi-weather-station-arduino-cc3000/arduino-sketch>. However, the wifi module used in this tutorial (TI CC3000) is out of date. A better module might be the ESP-01 (available online).

Ideally the transmitter unit would be housed in a stevenson screen (<http://www.karl-heinz-graf.com/stevensonscreen.pdf>), however I probably won't have time to make one. Also see <http://www.loganvillageweather.com/station/stevenson.html>.

Originally I wanted to use analog panel meters to display the information as this would provide a clear and low-power way of displaying the information. However, I have learnt that these panel meters are difficult to interface with and require a formula to convert the binary values to an analog voltage signal, as well as scales to go behind the meters. I have decided to use an LED display like the one in the Instructables tutorial (above).

I will power the external unit by a lead-acid 12V motorcycle battery I already own (connected to pin 30) which will be charged by a solar panel (already own). The receiver unit will be powered through the 5V mini-USB connection by a USB AC adapter I already own. I plan on housing the external electronics (excluding sensors) inside a plastic instrument case from Jaycar/Altronics (and ideally also in a stevenson screen), however I will buy this when I know the layout of components. I will house the receiver unit in either a wood case that I make or a plastic instrument case.

PARTS LIST

- [\\$12.50 x 2 Arduino Nano R3](#) (one tx, one rx)
- [\\$12.00 AM2302 Temp/Humidity sensor module](#) (based around DHT22) ([datasheet](#))
- [\\$5.99 LED Display](#) with HD44780 ctrl board for Arduino (20x4 char)
- [\\$10.00 RF tx/rx for Arduino](#) (433 MHz)
- [\\$8.00 BMP180 Pressure Sensor \(barometer\) & breakout board](#) ([datasheet](#))
- [\\$8.50 PIR motion sensor for Arduino](#) ([datasheet](#))
- [\\$14.95 Breadboard](#) (830 holes)
- \$4.99 instrument case for receiver
- \$3.99 instrument case for transmitter

Total - \$93.42

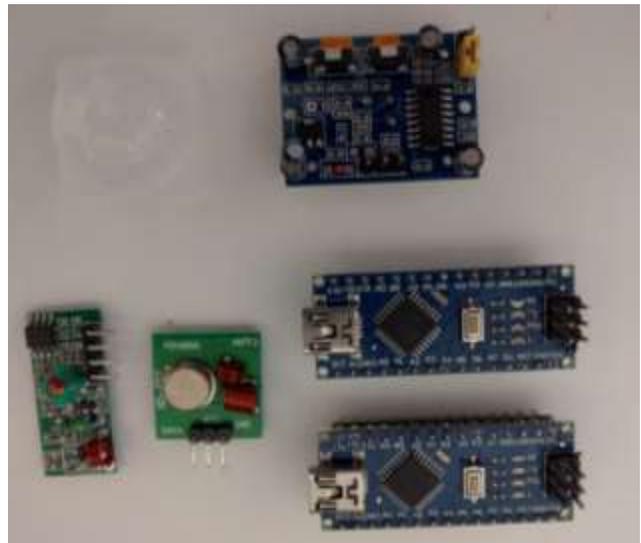
BUILD LOG

Much of Term 1 was spent researching and planning. I wanted to order the right parts the first time and be confident that they would work together, so I spent quite a lot of time reading datasheets, tutorials, and comparing electronic components. Once I ordered the parts they took about a week to be delivered, so during this time I researched further into programming the Arduino Nano and I developed a list of libraries that I would use.

Once I received the parts for my project in early March, I started by doing some tests with one Arduino board and the DHT22

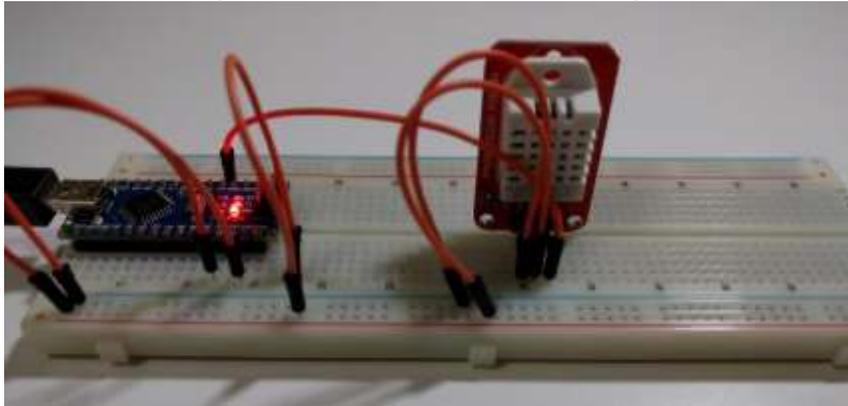
temperature/humidity sensor. It took quite some time for me to become familiar with the Arduino IDE as well as the programming language (a modified version of C), however with the help of examples I developed some working proof-of-concept tests. One problem that I experienced was finding libraries that worked with the Arduino Nano. Many sensor libraries (and their corresponding examples) are developed for other Arduino boards, or other processes (my boards use the more powerful ATmega32 processor). However, I got around this problem by doing lots of research to find a combination of libraries and examples that worked for my hardware configuration.

Towards the end of Term 1 I added the BMP180 pressure sensor onto the breadboard and started doing some tests. I experienced similar problems that I did with the DHT22, but I quickly developed some working code.



Thursday 21st April

Today I started work on adding the 433 MHz transmitter and receiver. Wiring up the devices which took some time as the pins on mine are labelled differently to most. I was then able to test them with examples



The breadboard with the Arduino and humidity sensor

from a few libraries.

Unfortunately none of the libraries I tried worked straight away, returning constantly high signals around the 350-400 range. It is possible this is from interference from nearby 433 mhz devices, but I doubt this is the case.

Friday 22nd April

I found that unplugging the 433 mhz transmitter does not change the readings, so this suggests they are caused by interference or bad coding on my part. Probably the latter.

Wednesday 27th April

Instead of trying to code everything from scratch, I tried modifying the code from the Instructables tutorial mentioned in the original plan, due to the errors I had been having. I'm sure I can fix these errors but given the time restraints I think it is best to build upon what is already available. After slightly modifying the wiring to suit the new code, I was able to add code for the BMP180 sensor (the original code is for a BMP085 - slightly different), as well as the DHT22 (originally DHT11). I encountered some minor errors in this process but they were fixed easily. Once I had made these changes I tried uploading the code to the Arduino Nano board, however I was faced with a cryptic error saying

```
"I:\\libraries\\OneWire.cpp I:\\libraries\\OneWire.h: The filename, directory name, or volume label syntax is incorrect. Error compiling."
```

Through some googling I found this was caused by storing the libraries on a USB drive (I did this so that I can work on the project at school). Apparently this is a limitation of the Windows kernel. Not sure why it doesn't happen for any other libraries as I am using >5, however I was able to fix it by temporarily storing the code and its libraries on my computer.

After this I experienced an error with avrdude (part of the Arduino compiler), which was caused by setting the wrong COM port. This was difficult to determine since the COM ports do not report the device connected. This was easily fixed.

Friday 29th April

Today I finished cleaning up the code. This is quite a laborious task because to start with the code was largely a combination of example programs. I spent several hours simplifying and commenting the code to make it easier to debug.

Sunday 1st May

Today I added the LCD display, which was not too hard. The display has 16 pins, however I am only using 11 of them. The display took some time to wire up, and I tried creating a circuit diagram in Fritzing to keep track of all the wiring. However, this was taking too much time and I ended up just drawing it on paper.

When I first connected the display the backlight was not on, and the text was very faint on the display. This was fixed by swapping two pins which the original tutorial had put the wrong way around.



The breadboard with LCD display added

Sunday 8st May

After finishing the coding, I split the transmitter and receiver segments into separate breadboards. I planned to test the range of the transmitter/receiver pair, but when I turned on the two Arduinos the receiver did not display the readings. This was an incredibly frustrating problem which I spent about a week trying to fix.

I did not know whether the problem was in the transmitting or receiving unit, nor whether it was a software or hardware problem. Looking back on the problem, the use of an oscilloscope would have helped greatly to determine the source of the problem. Eventually I found that the problem was caused by not setting the correct pin mode on the receiving Arduino (can be OUTPUT, INPUT or INPUT_PULLUP). The Arduino defaults all pins to Input, so I am not sure why it was set to output seemingly at random. It is possible that it was caused by accidentally shorting two pins of the Arduino, however I think this is an unlikely cause. More information about pin modes here - <https://www.arduino.cc/en/Reference/PinMode> and <https://www.arduino.cc/en/Tutorial/DigitalPins>.

Tuesday 10th May

After fixing the very frustrating problem caused by not setting the pin mode, I finalised the code and tuned the motion sensor (it has trimpots for sensitivity and duration). I also soldered antennas onto the transmitter and receiver boards.

After this I tested the range of the wireless signal, and I found it could comfortably go 20 metres through a few walls. At the edge of its range not all three values are received every time, but this does not matter as long as they are updated most of the time.

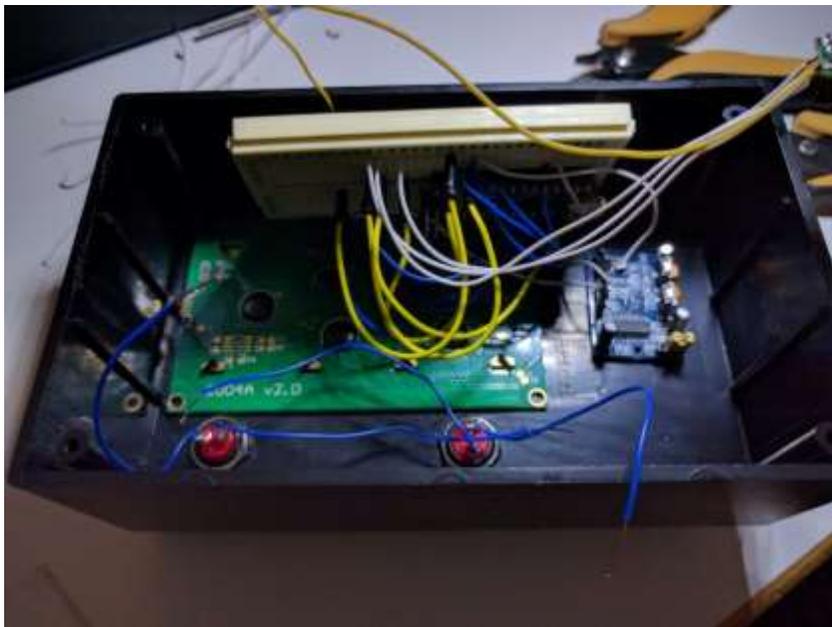
Wednesday 11th May

After taking the project to school I found that I experienced a problem with similar symptoms to the one previously mentioned, where the receiver would not receive any signals. I have eliminated any problems in the code as it did not change since it was working, but I am thinking either the transmitter or receiver module has failed. I am yet to fully test these things.

Saturday 14th May

After a very long time spent on investigating this problem, I found that it was caused by a bad connection from the transmitter module to the breadboard. The two smaller breadboards I purchased were very difficult to plug pins into and were not very reliable. I fixed this problem by soldering wires onto each module and plugging these into the breadboard as wires seem to work better.

Sunday 15th May



Today I started work on the two enclosures at home. A few weeks ago I bought two ABS boxes from Jaycar which will house the components. I have spent some time measuring each component with callipers and creating a template in Photoshop, which I will print out and stick over the boxes so I can drill and cut everything in exactly the right spot. The template is now finished and I am ready to start drilling.

Inside the receiver enclosure

Tuesday 17th May

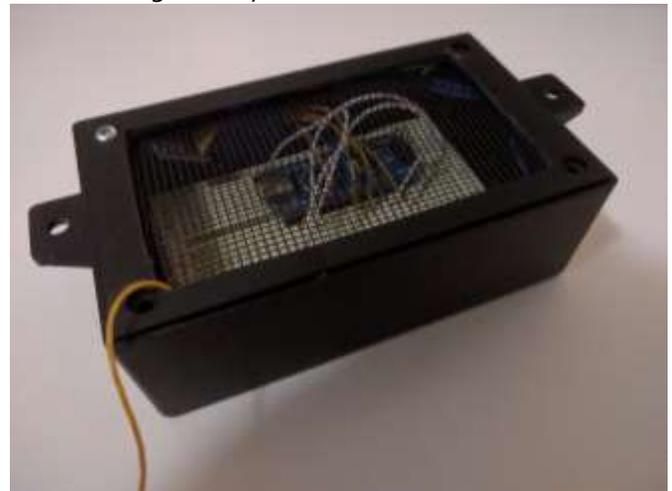
Today I finished the housing for the receiver unit. I used a drill press to drill all the holes, and a coping saw to cut out the section for the LCD display. I am very happy with the way the box looks, and it is better than I expected. I also wired up two pushbuttons to act as minimum and maximum buttons. This was very simple to add, and they simply display the min or max temperature since the device has been on. The min/max readings are shown on the display for 7 seconds, then the live readings are returned.

Wednesday 18th May

Today I worked on the transmitter housing. I cut a hole in the bottom of the box and glued flywire in here to allow proper airflow. If I had time I would build a miniature stevenson's screen (as mentioned in the original plan), however I think the box will provide sufficient weatherproofing and airflow. I also glued all the components into the box and routed the USB power cable through the flywire.



Completed receiver



Completed transmitter

PART B – EVALUATION QUESTIONS

1. What would you most like me to notice about this work?

I would most like you to notice how clean and concise the code is for the Arduinos. I spent a long time writing the code to make it concise, clean and functional. I believe writing good code is quite an art, and it is very important especially if I am sharing my code with other people. There are many intricate calculations and conversions that must be made in by both the receiving and transmitting unit, and I have done these in a concise but easy to understand way. The code is written in the Arduino language which is based on the language C.

I have uploaded the code for my project to GitHub, a code sharing platform, at <https://github.com/alorimer/arduino-weather>. It is publicly accessible (open source) so anyone else can use my code for their own purposes. More details about the code are visible in `readme.md` in the GitHub repository.

2. How well organised were you? How effective was your long term planning?

I am happy with my organisation and time management skills on this project. I have finished the project with time to spare, and I was not rushed in any part of the project. I spent time working on the project most days of the week, including at home when I didn't have a lesson at school. For example, I bought the enclosure for the project as soon as I knew what components had to fit in them. If I had bought them any sooner they would have likely been the wrong size, and any later I wouldn't have finished the project on time. I used a Gantt chart to plan my time which was a big help to make sure I was always on track. At times I did get behind as things took longer than I had anticipated, but I always compensated for this by working efficiently.

I am also proud of how I planned this project. I spent a long time considering different options for components, configurations and designs, and I think this paid off and helped to make my project successful. I believe most of the decisions I made before I started the project were crucial to its success and if I had made them during the project it would have been much harder. For example, at the start of the planning process I wanted to implement wifi instead of the 433 MHz wireless system. However during planning I found that this would take a lot more work and I would not finish it in time. This research and planning helped me to use my time efficiently and finish the project with time to spare.

My notes on planning are on page 1.

3. Creative solutions and problem solving

The project required many creative solutions and lateral thinking. For example, when the receiver module was not receiving any data I did not rule out anything until I could be absolutely sure. I did a few simple tests with the code which suggested the problem was caused by hardware. Many people would continue to search for problems in the hardware, but I did not rule out a problem in the code. After a while I found that indeed the problem was in the code, but if I had not been thinking laterally and with an open mind I would have ruled out any problems in the code.

I have solved a huge number of problems during this project – it is not like just putting together a kit where everything has been done before. Some of the problems were very small ones, for example most times I compiled the code there was at least one error with it. Each time this happened I had to use creativity, lateral thinking and research to solve these problems. The same is true of larger problems, such as the two times the receiver module was not receiving any data. I designed at least 20 tests involving hardware and software, and lateral thinking and creativity also helped a lot to solve these problems.

4. Suggestions for improvement

- Better accuracy of measurements – could be done through calibration and compensation
- More attractive enclosure for receiver, made from wood (I did not have enough time to do this)
- Transmit data through wifi and create a web application to generate graphs and log the data
- Neater cutting and drilling in the enclosures
- Attach proper connectors to the outside of the enclosures so that the cables are detachable

- Add a backup battery to the receiver unit so it stays on for some time when unplugged
- Use analog panel meters instead of an LCD display
- Add an anemometer to measure wind speed
- Put transmitter unit in a Stevenson screen for better weatherproofing and airflow

5. What was the greatest challenge?

The greatest challenge of this project was fixing the problem where the receiver module was not receiving any data. This was a mysterious problem which took about a week to solve, and it was very frustrating. Fortunately, it was a simple fix which I worked out with the help of the internet and some other people knowledgeable in digital electronics and development boards such as the Arduino.

6. What has been your best part of the project?

As described in question 1, the best part of this project is the coding. It is clear and concise whilst still being very functional. The entire system relies on clean, low-latency code – it could still work without a neat enclosure or tidy wiring, but the code is absolutely crucial to the function of the project.

The code for my project is visible on GitHub at <https://github.com/alorimer/arduino-weather>.

7. How did you respond when things went wrong?

I was always calm and persistent in the face of adversity, despite how frustrating and annoying the problems I experienced were. I never gave up, and I could always see “the light at the end of the tunnel”. For example, I spent a week trying to fix a problem with the wireless link. I never lost hope that I would fix this problem, and I knew from experience that it is possible to fix practically any problem I encounter. It was for this reason that I was able to persist until I fixed the problem, and this persistence helped me through all the challenges I faced throughout the project.

8. How do you think your fellow class members would rate your effort?

I think they would rate my effort about 9/10. I put a lot of effort into this project and this shows in the end result.

9. What kinaesthetic skills have you developed or improved?

I have improved mostly my fine motor control through soldering and other small work on PCBs and breadboards. I have also developed this skill through precise drilling and cutting. However, this project has less of a physical aspect than many others as much of the project is planning and coding.

10. Give specific examples of initiative

- Thinking of an original and interesting project idea that I am motivated about
- Researching on the internet extensively
- Asking on internet forums
- Talking to family members about problems I experienced
- Thinking of ways to solve problems and thinking laterally, without invocation from others

11. What have you learnt from this project?

- This project has given me a lot of experience in solving problems with digital electronics and developments boards. These skills I have developed will help me in the future.
- Almost any problem can be solved if you work at it intelligently and patiently
- If a problem seems like it is coming from a particular source, it is possibly something else that is causing it to seem like that. Electronics is deceptive.
- Breadboards can be unreliable. Use good quality ones that work with pins as well as wires.
- It is very important to know why something works before it stops working. Try to break your creation (within reason) to understand better how it works and its limitations.

12. Where do you need to improve?

- Problem solving quickly – many of the problems I encountered took a long time to solve, and whilst this was not a problem in this project I would like to improve on this.
- Cutting and drilling precisely and neatly
- General understanding of the Arduino language and compiler
- General understanding of digital electronics, particularly the Arduino

13. How do you feel about your project?

I am very happy with the way my project has turned out. I really enjoyed the learning experience and I have created a practical and interesting end product. I have learnt a lot more than I expected, and it has been a challenging but fun process.

14. What could you do next?

- Better accuracy of measurements – could be done through calibration and compensation
- More attractive enclosure for receiver, made from wood (I did not have enough time to do this)
- Transmit data through wifi and create a web application to generate graphs and log the data
- Attach proper connectors to the outside of the enclosures so that the cables are detachable
- Add a backup battery to the receiver unit so it stays on for some time when unplugged

15. List 4 things that you have learned while completing this project

1. Almost any problem can be solved if you work at it intelligently and patiently
2. It is very important to know why something works before it stops working. Try to break your creation (within reason) to understand better how it works and its limitations.
3. If a problem seems like it is coming from a particular source, it is possibly something else that is causing it to seem like that. Electronics is deceptive.
4. The best way to check digital electronics is to write down the voltage each pin should have, and then test each pin with a multimeter or oscilloscope.
5. Breadboards can be unreliable. Use good quality ones that work with pins as well as wires.

PART C – ANIMATED GIF

Here is a link to a GIF showing the inside of both units - <https://gfycat.com/SpiffyMildBlackbird>