The idea for the plant monitor started when my wife said: I like plants, but I have no idea when to water them. The lightbulb went off. I had an Arduino and some LED’s, I only needed a soil moisture sensor, thankfully they’re inexpensive.

Once I had the parts collected, I set everything up with an Arduino Uno. And after a little research regarding the best minimum volumetric water we had an LED notifying us when the plant needed watering. Iteration one was basic and ugly (not that any of my end products are very elegant). Iteration two included a [Latte Panda](https://www.dfrobot.com/product-1585.html) as a way to send the data to the internet. I went through the process of defining a database in Azure that would host my information. I wrote a C# .NET serial monitor tool that would gather the information from the onboard Arduino and push it to my database. From there I went nuts. I built out a .NET API app service as well as a .NET MVC Angular website that displayed the basic information using high charts. The app service took the historical information and gave historical watering information, as well as a projected water date based on the trends. On top of that I decided to build a *very* basic android application showing similar chart data. I started working on push notifications, but abandoned that because my wife has an iPhone and I don’t want to pay for a developer license. While those shenanigans were happening, I did setup the Arduino Uno to monitor four of our plants in a home-made box with the notification lights only. This meant that the data on the site was only for one plant, though it was written dynamically to accommodate multiple plants.

I decided I wanted to clean up my azure instance and wrap up the online portion of the plant monitoring application, I had done all I wanted and learned a lot, thus I felt good wrapping things up, though my code could have had a few instances of refactoring.

That isn’t the end of the plant monitor. I knew to myself there had to be a less costly way than a $200 windows machine with Arduino built in. After a summer and fall of not investing time in this project I decided I wanted a weather station to correlate the plant water consumption with temperature, humidity, and barometric pressure. Upon research of a cost effective weather station I discovered the [ESP8266](https://www.espressif.com/en/products/hardware/esp8266ex/overview) I specifically purchased the 12-E model. With the weather station kit costing a whopping $20 I started digging in again. I first followed some of their tutorials to get a grasp on writing to the ESP and sharing the data via an API. I proceeded to write my own version using the Arduino IDE and thingspeak as the API and data host.



Now again, I’ve got the data of our Pony Tail palm tree online, and this time I have weather data so I can map and correlate trends.  
  
Check out the chart on my other page.