

Saving and Transmitting Data Using the Arduino YÚN

ECE 480 Design Team 3

Jacob Jebb

11/11/2015

Executive Summary

As technology has advanced, we are able to create more versatile systems that can generate large amounts of data. With this data we can find trends and other events that will explain what exactly is happening within a particular system. The Arduino YÚN offers cost effective way to log data and access this data without ever entering a control panel. Using the internet and basic programming knowledge it is possible to setup the microcontroller to save data to an onboard micro-SD card and to transfer that data to a computer.

Keywords:

Introduction

Having a wide variety of options for choosing a microcontroller, the Arduino YÚN was selected because of the functionality it can provide without purchasing any other shields and ease of programming. The YÚN comes packaged with both Ethernet and Wi-Fi internet connection. We choose to use the Ethernet connection, to have a more reliable connection. The YÚN also has an on board micro-SD card slot. All the connections to the YÚN can be seen in Figure 1.

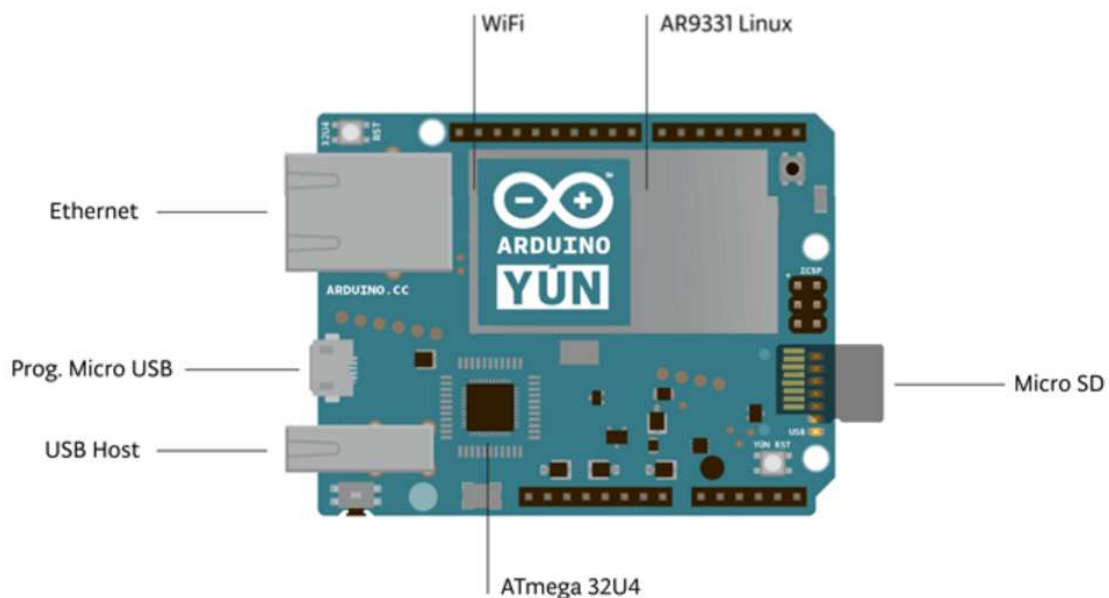


Figure 1. Arduino YÚN

Data Logging

To save the data to the micro-SD card a folder named “arduino” and a text file named datalog.txt must be added to the root directory of the SD card. From there we can use the following code from the Arduino website:

```
#include <FileIO.h>

void setup() {
  Bridge.begin();
  Serial.begin(9600);
  FileSystem.begin();

  while(!Serial);
  Serial.println("Filesystem datalogger\n");
}

void loop() {
  String dataString;
  dataString += getTimestamp();
  dataString += " = ";

  for (int analogPin = 0; analogPin < 3; analogPin++) {
    int sensor = analogRead(analogPin);
    dataString += String(sensor);
    if (analogPin < 2) {
      dataString += ",";
    }
  }

  File dataFile = FileSystem.open("/mnt/sd/datalog.txt", FILE_APPEND);

  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    Serial.println(dataString);
  }
  else {
    Serial.println("error opening datalog.txt");
  }
  delay(15000);

  String getTimestamp() {
    String result;
    Process time;
    time.begin("date");
    time.addParameter("+%D-%I");
    time.run();

    while(time.available()>0) {
      char c = time.read();
      if(c != '\n')
        result += c;
    }

    return result;
  }
}
```

The code allows us to take the signal coming into the analog pins and save the data points to SD card. We are also able to save the data with a time stamp, the time stamp gives a way to compare data with a certain event they may have happened.

Internet Connection

Before we can start transferring the data from the YÚN we need to set-up the Ethernet connection. The first step is downloading the OPENWRT upgrade image from the Arduino Software website. This image is installed onto the SD card. We then need to connect to the YÚN's Wi-Fi signal seen in Figure 2.

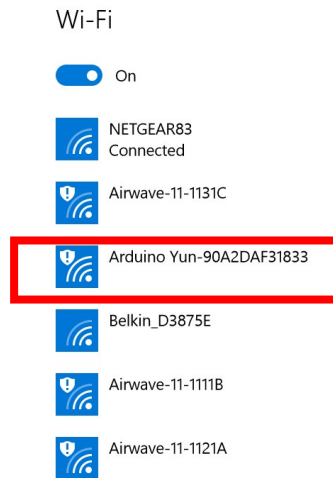


Figure 2. Wi-Fi Connections

Once connected to the YÚN we can open a web browser and go to arduino.local/. From there we insert the default password: arduino and we are now able to get into the settings of the YÚN. To start we needed to update the OPENWRT software on the microcontroller, as you can see in Figure 3, we need to click the reset button under the Image File.

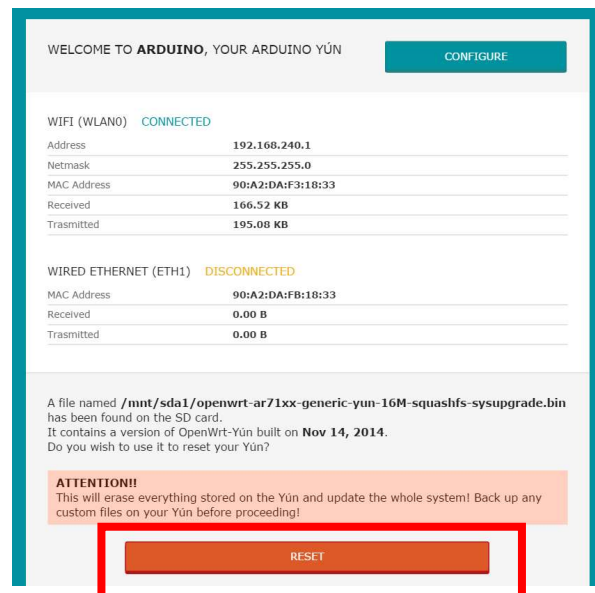


Figure 3. Setting Page

We can now use a SSH program, such as PUTTY, to remote into the microcontroller. Using the specific IP address of this particular microcontroller. We will have the following setup:

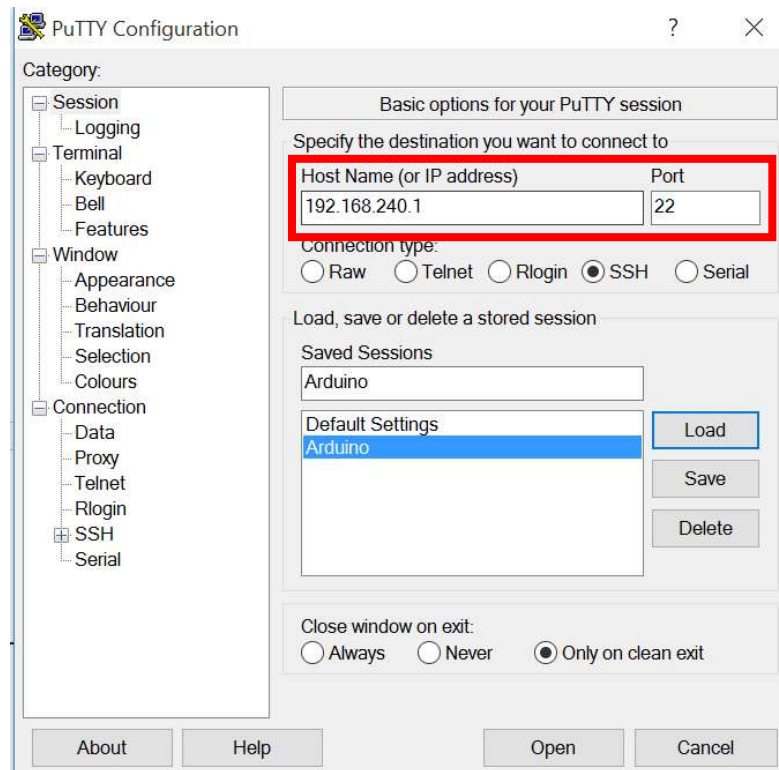


Figure 4. Putty Settings

Before we can install the OPENWRT packages we need to restart the network to make sure we are using the correct settings. Using the route command we can see the Kernel IP routing table. We then need to use the /etc/init.d/network restart command, to be able to download are packages. Figure 5 shows the results of these commands.

[illegible]

Figure 5. Kernel IP Routing Table and Network Restart Command

We are now able to install the necessary OPENWRT packages that will allow us wirelessly transfer the datalog.txt file from the microcontroller to our computer. The package manager is an important tool for Linux systems, it allows for a simple way to install, update and remove pre-configured applications. The package manager needs an updated database to display the available packages for the microcontroller, we can do this by running the `opkg update` command. This database is then saved to the RAM, which means that if you want to install other applications the `opkg update` command needs to be run after any reboot. We are now ready to install `vsftpd openssh-sftp-server` package. This application will allow us to remote into the microcontroller and transfer the text file to our computer.

```
root@Arduino:~# opkg update
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/Packages.gz.
Updated list of available packages in /var/opkg-lists/attitude_adjustment.
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/Packages.sig.
Signature check passed.
root@Arduino:~# opkg vsftpd openssh-sftp-server
```

Figure 6. *Commands to Install Package*

After the package is installed onto the microcontroller we can install a FTP (File Transfer Protocol) software to the desired computer. This will allow us to take our data log text and transfer out to our computer. After researching the FileZilla offers a user friendly interface to allow us achieve our goals. Once FileZilla is installed we can setup the saved site to connect to, so we won't have to insert the settings each time we want to access our text file. To do this we need to go to File -> Site Manager. We can then add a new site named Arduino Yun. We will then add the following data found in figure 7.

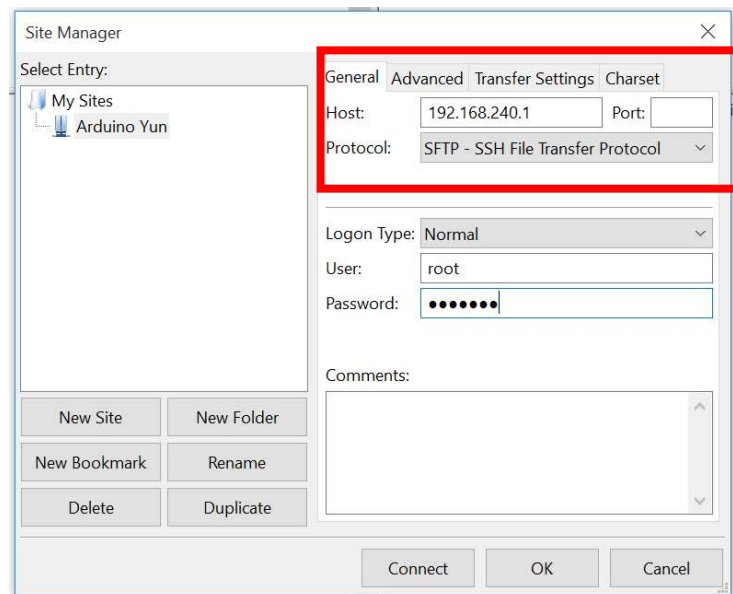


Figure 7. Settings to Connect to Microcontroller

Now that we are connected to the microcontroller, all that is left to do is insert the correct file path and then transfer the data. In the remote site we want to insert the following path: /mnt/sda1. This will take us to the micro-SD card's root directory. We can then either right click on the text file and choose transfer or double click on the file to transfer to the computer.

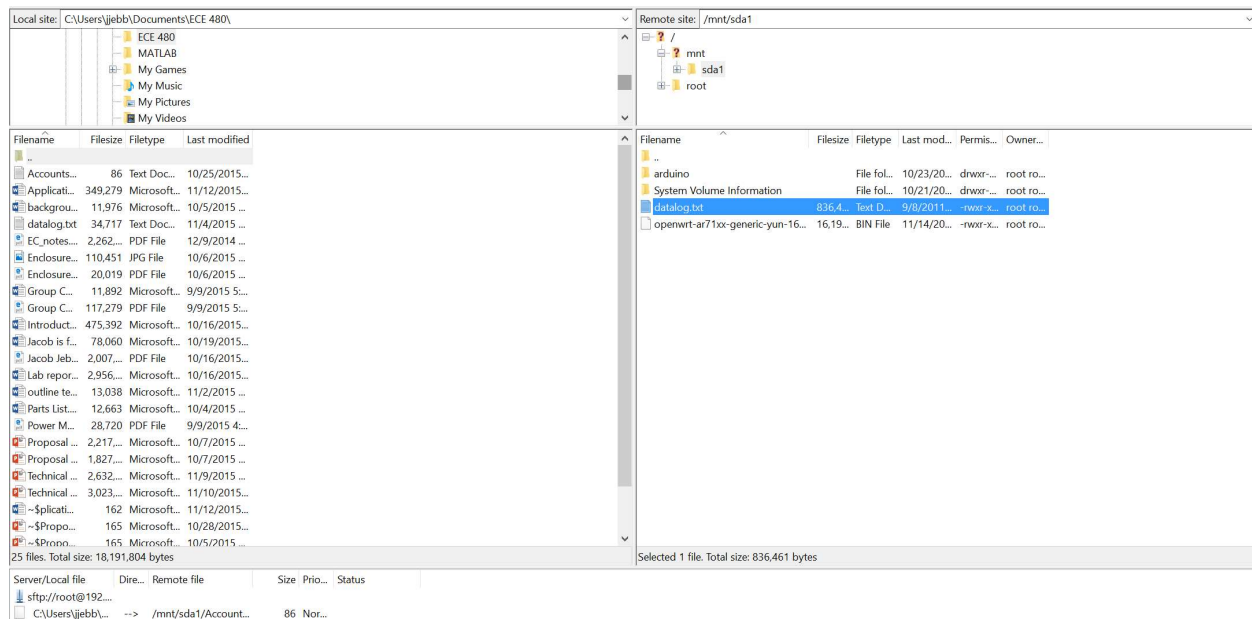


Figure 8. Transferring the Text from the Microcontroller to Computer.

There is also a possibility to change where the file is going to be stored on the computer. We can also take files from the computer and upload them to the microcontroller. This package and software offers a safe way to extract data without ever having to enter the panel.

References

<https://www.arduino.cc/en/Main/Software>

<https://www.arduino.cc/en/Tutorial/YunDatalogger>

<https://filezilla-project.org/>