



DREXEL UNIVERSITY

Senior Design

Electrical and Computer Engineering

USER MANUAL

Team Number ECE-10

Solar Power Forecasting Tool

Team Members

Name

Kim Nguyen

Kara Ogawa

Stephan Tang

Department

ECE

ECE

ECE

Email

kn383@drexel.edu

kao73@drexel.edu

st643@drexel.edu

Team Advisor

Name

Dr. Niebur

Department

ECE

Email

niebur@coe.drexel.edu

Date : 13 May, 2017
Revision : 00
Prepared by : ECE Group 10: Kim Nguyen, Kara Ogawa, Stephan Tang
Distribution :

Table of Contents

1. Introduction	4
1.1. Document Identification	4
1.2. System Overview	4
1.3. Document Overview	4
1.4. Reference Documents	4
1.4.1. Acronyms and Abbreviations	4
2. System Description	4
2.1. Introduction	4
2.2. Operational Scenarios	4
2.3. Conceptual Design	5
3. The Solar Array	6
3.1. Location	6
3.2. Technical Details	6
4. Data Acquisition Systems	7
4.1. Scope of Hardware System	7
4.2. Hardware Design	7
4.2.1. Power Supply	7
4.2.2. Script Design	7
4.3. Installation Details	10
4.3.1. Master DAQ	10
4.3.2. Slave DAQ	11
4.4. Hardware Validation	11
4.5. Hardware Calibration Procedures	11
4.6. Hardware Maintenance and Adjustment	11
5. Database	12
5.1. Logging In	12
5.2. Tables	12
6. User Interface Design	13
6.1. Classes of Users	13
6.2. Developer Files	13
6.2.1. Raw File Access	13
6.2.2. General Appearance	13
6.2.3. Webpages	14
7. Software Design	15
7.1. Algorithm Design Process	15
7.1.1. Model Testing for Selection	15
7.2. Software Design Description	15
7.2.1. Software Components	15
7.3. Preconditions for Software	15
7.3.1. Preconditions for System	15
7.4. Running the Software	15

1. Introduction

1.1. Document Identification

This user manual describes the design of the Drexel University senior design project, the Solar Power Forecasting Tool. This document is prepared by ECE Group 10 as a final deliverable.

1.2. System Overview

The objective of the senior design project is to develop an accurate solar forecasting algorithm as a method to determine future power output for the solar array installation at Drexel University. The system designed for this project consists of a user interface, hardware, and software components. This tool is a continuation of the 2015-2016 ECE Team 16 Senior Design Project, the Santino Solar Power Prediction Tool.

1.3. Document Overview

The purpose of this document is to provide a detailed guide for the operation of the solar power forecasting tool. It is to supplement the progress report, which describes the design and work completed on this project.

1.4. Reference Documents

The present document is prepared on the basis of the following reference documents, and should be read in conjunction with them.

- “Progress Report, Spring 2016-2017 Solar Power Forecasting Tool”. ECE Group 10, Philadelphia, May 2017.

1.4.1. Acronyms and Abbreviations

Table 1-1 lists the acronyms and abbreviations used in this document.

Table 1-1: Acronyms and Abbreviations.

Acronym	Meaning
DAQ	Data Acquisition System
SVMR	Support Vector Machine Regression Model
NARX	Nonlinear Autoregressive Exogenous Model

2. System Description

2.1. Introduction

The hardware system includes a Master and a Slave Data Acquisition Systems (DAQ). Each DAQ includes a Raspberry Pi and an Arduino which act as a communication between sensors and the web server. The Master DAQ contains a temperature sensor and an irradiance sensor (pyranometer) which measure temperature and solar irradiance around the PV panel on Drexel Main Building roof. The slave DAQ measures open-circuit voltage across the panel in Drexel Power Lab. This real-time data is sent via Drexel’s dragonfly3 Wi-Fi to a Drexel University database. The user interface is the website where this data is viewable. The software system includes the forecasting algorithm based on the support vector machine regression model (SVMR). It uses historical data from the National Renewable Energy Laboratory (NREL) to train the forecasting model and uses real-time data from the hardware system to forecast solar power up to a day ahead.

2.2. Operational Scenarios

This system is designed for daily use to collect real-time data and forecast solar power output up to a day ahead. Website users will be able to see forecasted weather data, real-time irradiance, temperature, and solar panel voltage, as well as forecasted solar power output.

Over time with regular training of weather data, the algorithm will be able to predict output power more accurately.

2.3. Conceptual Design

The overall breakdown of the hardware, server, and forecasting algorithm is shown in Fig. 1. These components are detailed in the following sections.

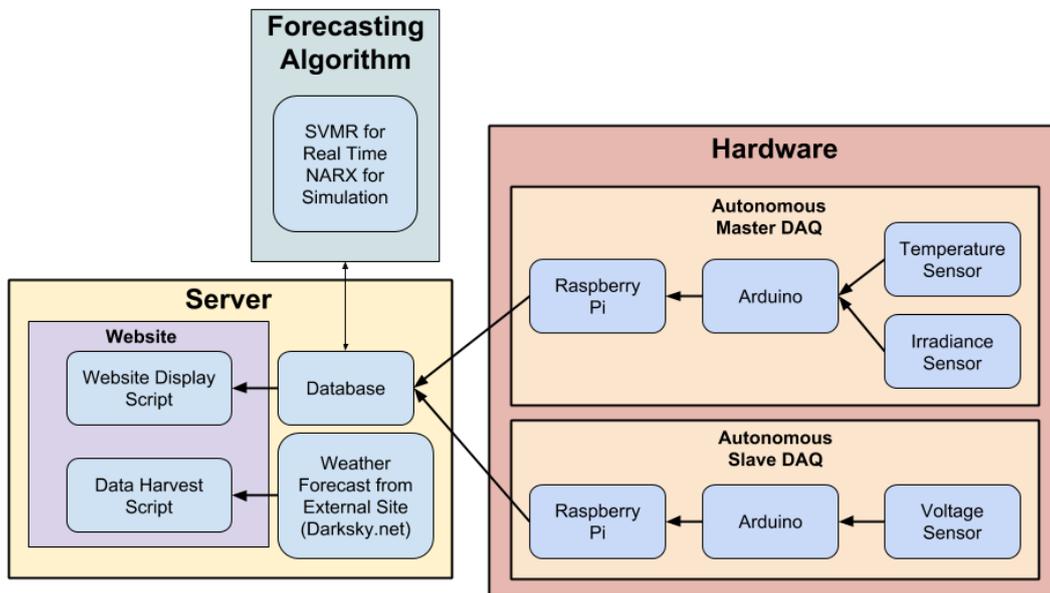


Fig. 1. Conceptual Design of System.

3. The Solar Array

3.1. Location

The Drexel University solar array was installed for research by the Center for Electric Power Engineering. It is located on the roof of the Main Building and is accessible with assistance from Drexel Facilities through a door on the 4th floor of the Main Building. Since there are only four facilities managers for Drexel University, advance scheduling for roof access is required. The team's contact was Jaime Barbaro at jb973@drexel.edu



Fig. 2. Solar array location.

3.2. Technical Details

The array consists of 8 Kyocera KD210GX-LP panels. The solar output DC power comes into the Drexel University power lab behind the transfer panel where the slave DAQ can be connected to read voltage. Either outlet beneath the solar panel as shown in red in Fig. 3 can be utilized to power the master DAQ.

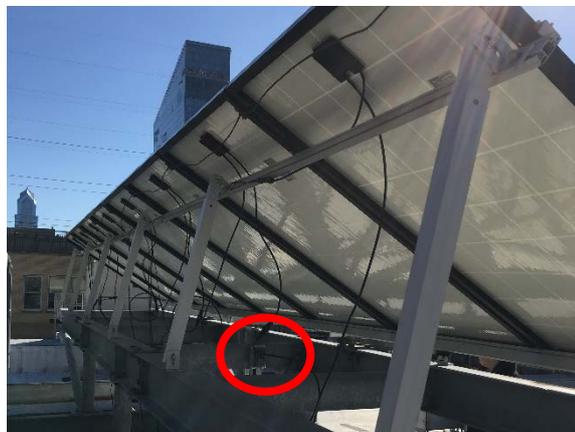


Fig. 3. Master DAQ location beneath array.

4. Data Acquisition Systems

4.1. Scope of Hardware System

The autonomous DAQ system consists of both a master DAQ at the solar panel on the roof of Main Building and a slave DAQ located in the Power Lab. The master DAQ consists of a temperature sensor and irradiance sensor connected to an Arduino. The Arduino collects the data from the sensors and directs the information to the Raspberry Pi through the Serial Port. The slave DAQ consists of an Arduino measuring voltage across the solar panel voltage terminals. This Arduino is also connected to a Raspberry Pi. Both Raspberry Pi's send data autonomously via Drexel Wi-Fi dragonfly3 to the pi_get.php script at http://solar.ece.drexel.edu/pi_get.php. The pi_get.php script organizes the data coming in asynchronously from the master and slave DAQs and sends it to the database. Information about accessing website files can be found in the User Interface Design section.

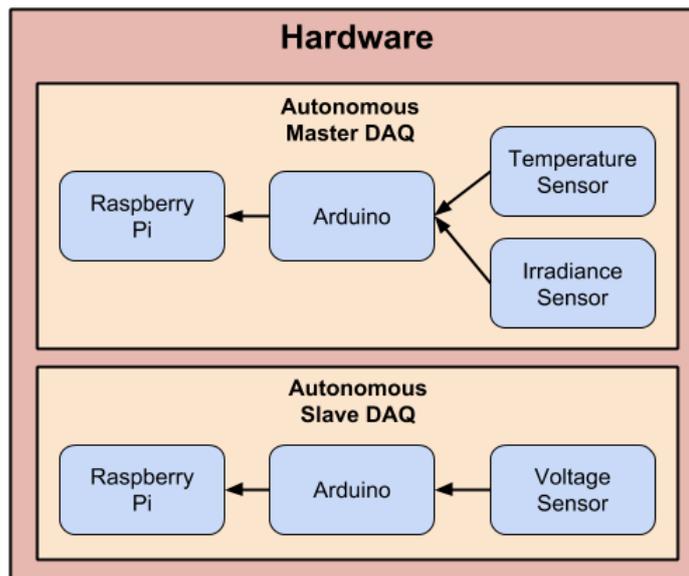


Fig. 4. Data Acquisition System Diagram

4.2. Hardware Design

4.2.1. Power Supply

Both the master and slave DAQs are powered through the Raspberry Pi, which requires a 5 V supply through a micro-USB. The power cord can be plugged into a standard U.S. power outlet.

4.2.2. Script Design

The master DAQ collects temperature and irradiance data in 5 minute intervals as value0 (PV temperature), value1 (irradiance), and value2 (die temperature) and sends these values to http://solar.ece.drexel.edu/pi_get.php. The PHP script connects to the database and populates a new row with these values as well as a value of -1 for the voltage value. The slave DAQ collects voltage data and also sends this value to http://solar.ece.drexel.edu/pi_get.php. The PHP script recognizes this data as voltage data and instead searches for the previous row with a value of "-1" to swap with the real voltage value.

4.2.2.1. Master DAQ

The weatherproofed master DAQ is shown in Fig. 5.

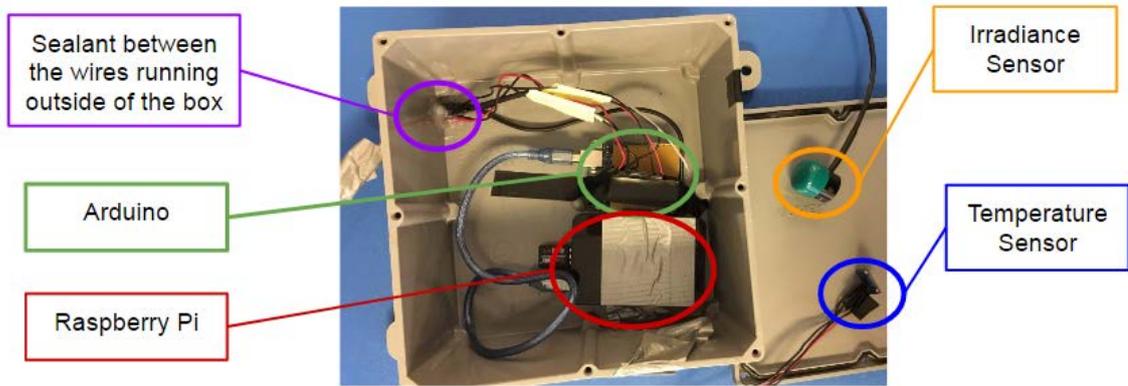


Fig. 5. Inside of the master DAQ.

The original circuit schematic for the temperature and irradiance sensors to the Arduino on the master DAQ is shown in Fig. 6.

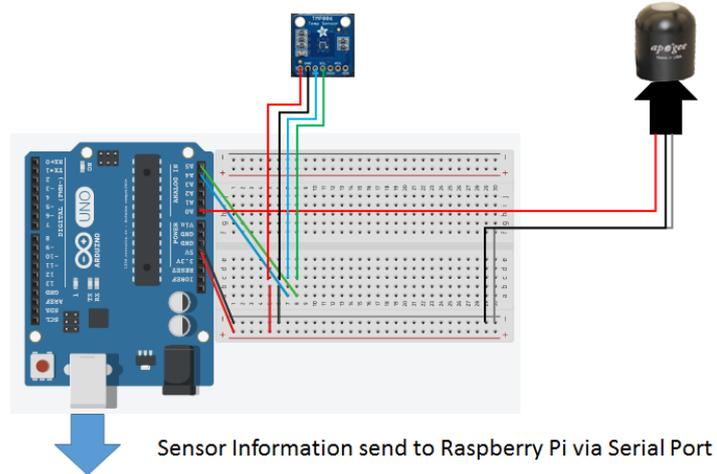


Fig. 6. Circuit schematic of master DAQ.

For greater durability, a printed circuit board was built. The schematic for the updated board is shown in Fig. 7 with the same connections of the temperature sensor at A4 and A5 and the irradiance sensor at A0.

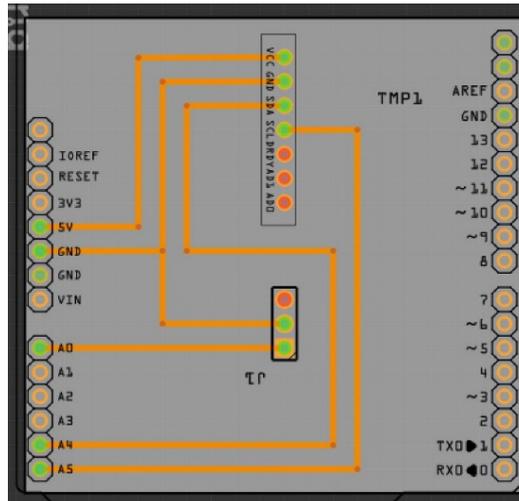


Fig. 7. PCB schematic for master DAQ.

4.2.2.1.1.

Temperature Sensor

The temperature sensor utilized is the Texas Instrument's TMP006, which detects infrared waves and measures temperature without contact. This sensor was chosen for its low cost and Adafruit support with an Arduino library. It was purchased from Adafruit already mounted on a breakout board.

Two temperatures are measured by the temperature sensor: the PV temperature and the die temperature.

The guide from Adafruit is available at:

<https://learn.adafruit.com/infrared-thermopile-sensor-breakout/overview>

4.2.2.1.2.

Irradiance Sensor

The irradiation sensor utilized is the Apogee Instruments' SP-110 Pyranometer. It has a silicon cell photodiode and is fully weatherproof.

4.2.2.2. Slave DAQ

The circuit schematic for the voltage sensor to the Arduino on the slave DAQ is shown in Fig. 8.

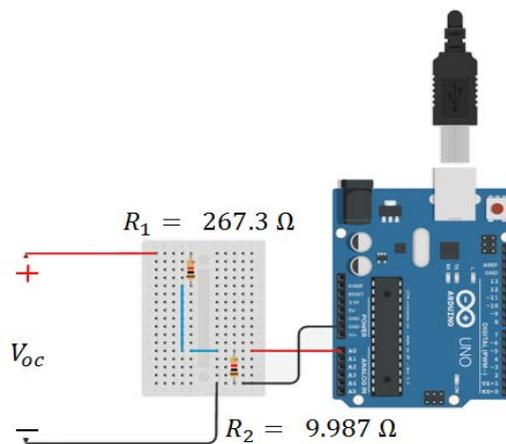


Fig. 8. Circuit schematic of the slave DAQ.

4.2.2.2.1. Voltage Measurement

A breadboard with a 267.3 Ω resistor and a 9.9 Ω resistor in series is connected to the solar array terminals in the Power Lab. The Arduino measures a portion of the open circuit voltage across the 9.9 Ω resistor. This resistor is in series with a 267.3 Ω resistor on a breadboard to limit voltage seen at the Arduino input.

4.3. Installation Details

4.3.1. Master DAQ

The master DAQ was installed on the roof of Main Building beneath the solar panel. The irradiance sensor was installed 1.5 inches beneath the array and angled with the panel tilt as shown in Fig. 9.



Fig. 9. Irradiance sensor mounted.

The temperature sensor was mounted on a pole 23 inches beneath the array and its edges were covered with electrical tape to further weatherproof it as shown in Fig. 10.



Fig. 10. Temperature sensor mounted.

The full set-up is shown in Fig. 11.

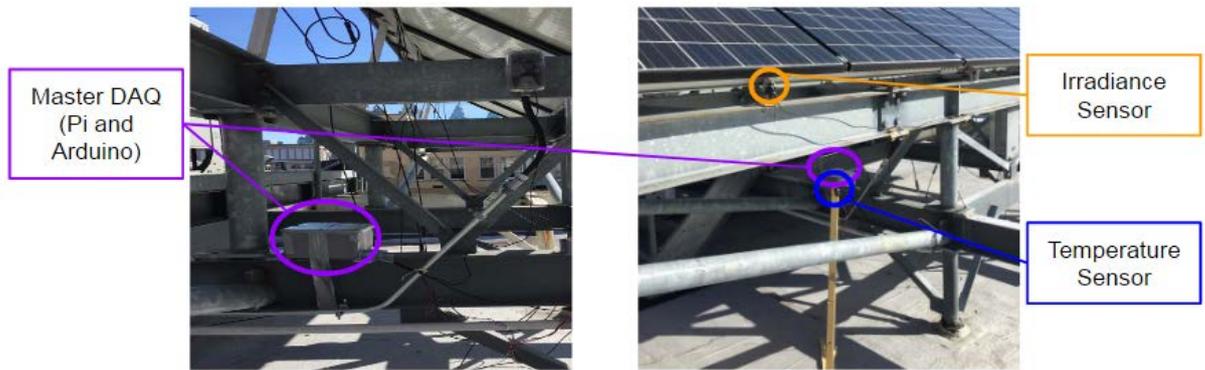


Fig. 11. Master DAQ set-up.

4.3.2. Slave DAQ

The Slave DAQ was installed in the Power Lab at the terminals behind the transfer panel as shown in Fig. 12. The Arduino is not pictured.

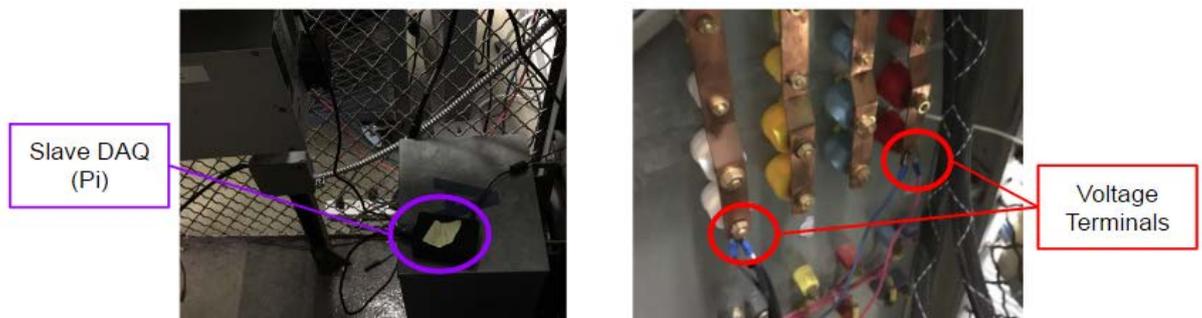


Fig. 12. Slave DAQ set-up.

4.4. Hardware Validation

The temperature measurements can be validated by comparing the measured temperature data to known temperature data.

The irradiance measurements are difficult to validate as irradiance data is not readily available. However, the irradiance sensor is pre-calibrated by Apogee Instruments and measurements can be compared to previously observed values.

The solar panel voltage measurements can be validated by connecting the slave DAQ to a power supply and comparing the measured voltage to multimeter measurements at the terminals. These were within a 5 V range of the multimeter.

4.5. Hardware Calibration Procedures

No additional calibration is required once the hardware is installed.

4.6. Hardware Maintenance and Adjustment

Routine maintenance can be performed for both DAQs by checking sensor positions and reinforcing weatherproofing as needed. This is more important for the master than the slave DAQ as it is stationary on the roof of Main Building.

5. Database

5.1. Logging In

The database credentials can be found in the file “databasecredentials.txt” in the website PHP files folder. The database can be accessed from Drexel University computers or from personal computers via Drexel’s Virtual Private Network (VPN). Users requiring access to these files are advised to contact Andrew Marx/IRT at Drexel University Computing and Technical Services at: helpdesk@coe.drexel.edu.

Windows users: \\solar.ece.drexel.edu\Sites\Custom ECE Solar Senior Design

Non-windows users: smb://solar.ece.drexel.edu\Sites\Custom ECE Solar Senior Design

The database is accessible at: <https://tech.coe.drexel.edu/phpmyadmin/>

5.2. Tables

The info_storage2 table stores the real-time data sent from the DAQ system. This is displayed on the Ras-Pi Data page on the website.

The info_storage4 table stores the forecasted weather data queried from the DarkSky API. It is triggered by pressing the “Get Forecast” button on the Forecast page on the website.

6. User Interface Design

The server design, incorporating the website user interface, is shown in Fig. 13.

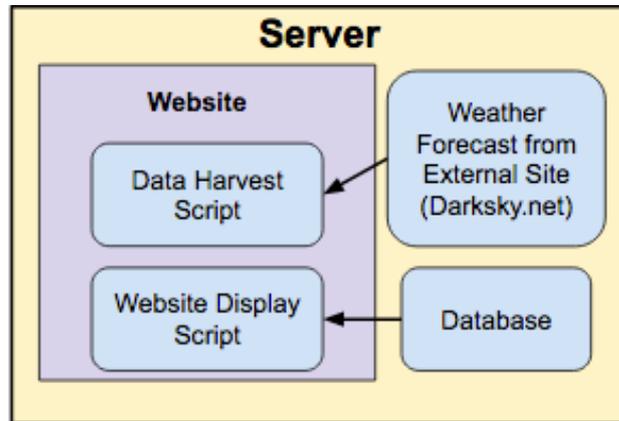


Fig. 13. Server Design Diagram.

6.1. Classes of Users

There will be one class of users viewing the website. All users will be able to access the forecasted weather data from the DarkSky API and the sensor data from the hardware DAQ system. The project website is: <http://solar.ece.drexel.edu/>

6.2. Developer Files

6.2.1. Raw File Access

All php files can be accessed from Drexel University computers or from personal computers via Drexel's Virtual Private Network (VPN). Users requiring access to these files are advised to contact Andrew Marx/IRT at Drexel University Computing and Technical Services at: helpdesk@coe.drexel.edu

Windows users: \\solar.ece.drexel.edu\Sites\Custom ECE Solar Senior Design

Non-windows users: smb://solar.ece.drexel.edu\Sites\Custom ECE Solar Senior Design

6.2.2. General Appearance

Files to update the header, footer, and pictures are found in the "includes" folder.

6.2.3. Webpages

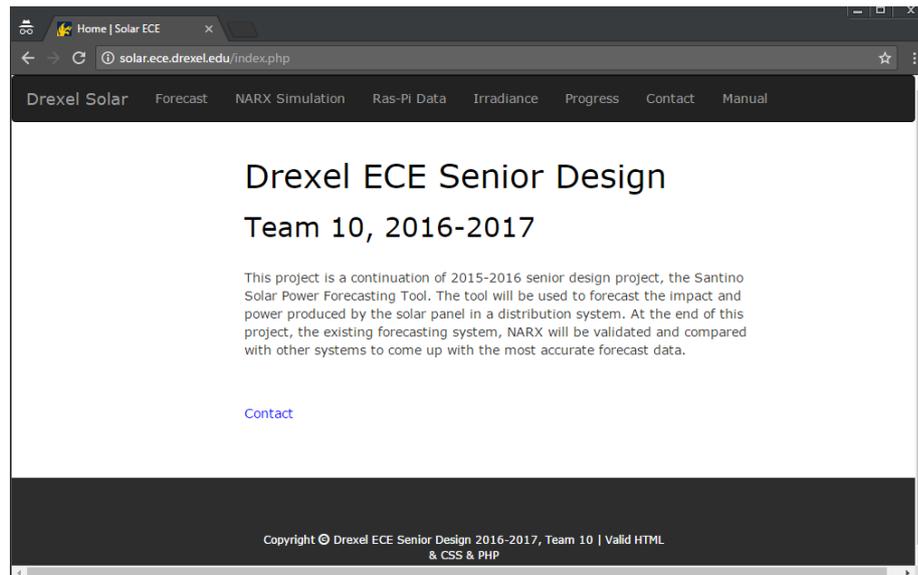


Fig. 14. Current project website.

6.2.3.1. Drexel Solar/Homepage - index.php

This is the homepage containing a brief description of the project.

6.2.3.2. Forecast - get_forecast.php

This forecast uses SVMR to forecast in real-time.

The "Get Forecast" button calls the weather forecast for the next 48 hours from DarkSky API and stores this information in the database. Request format information can be found here: <https://darksky.net/dev/docs/forecast>

After the SVMR forecast MATLAB file is run, the Irradiance and Power columns are updated with forecasted irradiance and power values.

6.2.3.3. NARX Simulation – NARX_simulation.php

This is a forecast simulation compiled with data previously collected using the NARX model to forecast. The forecasted data from DarkSky is shown alongside the data from the DAQs and the forecasted data based on the NARX model.

The NARX model was determined to be the best model but is unable to be implemented as the forecasting algorithm since the hardware is not functioning.

6.2.3.4. Ras-Pi Data - info_storage.php

The data from the database received from the DAQs is displayed with the forecasted irradiance and power based on the NARX model.

6.2.3.5. Irradiance - irradiance2.php

Irradiance (W/m^2) and voltage (V) measurements collected by the DAQs are plotted together.

6.2.3.6. Progress - progress.php

Progress on the senior design project is briefly documented here.

6.2.3.7. Contact - contact.php

This is a contact form that currently sends queries to kao73@drexel.edu. It should be updated to reflect the contact information of a current team member.

6.2.3.8. Manual

This is the location to download the most recent User's Manual.

7. Software Design

The software includes the SVMR model files that use the forecasted and real-time data to predict power from the Drexel University solar array. After the Master DAQ is repaired and the data collection is resumed, the NARX model should be used.

7.1. Algorithm Design Process

7.1.1. Model Testing for Selection

Data from the National Renewable Energy Laboratory (NREL) were compiled into yearly and seasonal data sets. Yearly data was divided into four data sets with three months of data to create a season (Fall, Winter, Spring, and Summer) to create the seasonal data sets. Input data for the training set included month, day, hour, temperature, windspeed, relative humidity, dew point, wind direction, and pressure. The comparison of the algorithm RMS train and test errors is shown in Table 2. The yearly NARX model was determined to be the best model. However, it is dependent on sensor data which is currently unavailable.

Table 2. Algorithm Error Comparisons

<i>Data Set</i>	<i>Model</i>	<i>RMS Error (Train)</i>	<i>RMS Error (Test)</i>
Yearly	SVMR Gaussian	0.1327	0.5524
Seasonal Average	SVMR Gaussian	0.1523	0.3526
Yearly	SVMR Linear	0.2458	0.2529
Seasonal Average	SVMR Linear	0.2314	0.2484
Yearly	NARX	0.1352	0.1794
Seasonal Average	NARX	0.3089	0.3198

7.2. Software Design Description

These components are all required by the MATLAB file, "SVMforecast.m" which is used to forecast solar irradiance, current, and thus output power.

7.2.1. Software Components

7.2.1.1. Model File - SVMmodel.mat

The SVMR model file

7.2.1.2. Database Communication - mysql-connector -java-5.1.41-bin.jar

The database communication between the MATLAB file, website, and database is established.

7.2.1.3. IV Power Calculation - iv_power_calculation.m, iadj.mat, vadj.mat

The IV curve from the Kyocera datasheet was digitized into the iadj.mat and vadj.mat files for current and voltage data points, respectively. The iv_power_calculation.m file plots these points and uses the measured irradiance to predict current. Current and voltage are multiplied to calculate output power.

7.3. Preconditions for Software

7.3.1. Preconditions for System

The user running the SVMforecast.m file must be on Drexel Wi-Fi to access the database during the forecast.

7.4. Running the Software

- 1) Select the "Forecast" tab on the project website, solar.ece.drexel.edu and press "Get Forecast." Wait until the word "Ready" appears on the bottom left of the page.
- 2) Run the SVMforecast.m MATLAB file. Wait until the "Forecast Complete" window appears in MATLAB.

- 3) Refresh the "Forecast" tab on the project website. The "Irradiance" and "Power" columns will be updated with the forecasted solar irradiance and power data.