MICHIGAN STATE UNIVERSITY | College of Engineering

ECE 480 - Design Team 2
Stu Andrzejewski, Kelly Desmond, David Rogers, Brad Garrod

Wireless Sensor Network Health Diagnostic

Facilitator : Dr. Tongtong Li (Michigan State University)
Mentor: Mr. Joseph Natarian (AFRL)
Technical Advisor: Mr. Kenneth Littlejohn (AFRL)

AFRL STUDENT CHALLENGE

Abstract

The United States Air Force relies on wireless sensor networks for protecting its equipment and U.S. military personnel around the world. Thus, it is paramount that these sensor networks are working properly. In order to ensure the integrity of a sensor network it must be monitored for potential failures. This project revolves around creating a health diagnostic for detecting sensor node failures across the network in real time.

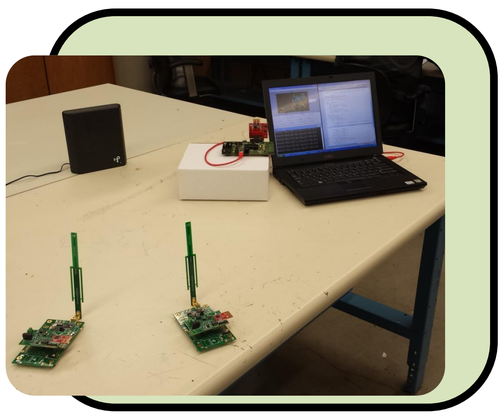
Project Overview

Problem Statement:
In order to ensure a properly functioning wireless sensor network, the Air Force needs a diagnostic tool that is easy to use, accurate and reliable.

Approach:
Configure a wireless sensor network, develop a graphical user interface, and design algorithms to analyze sensor metrics vital to network health.

Attack:

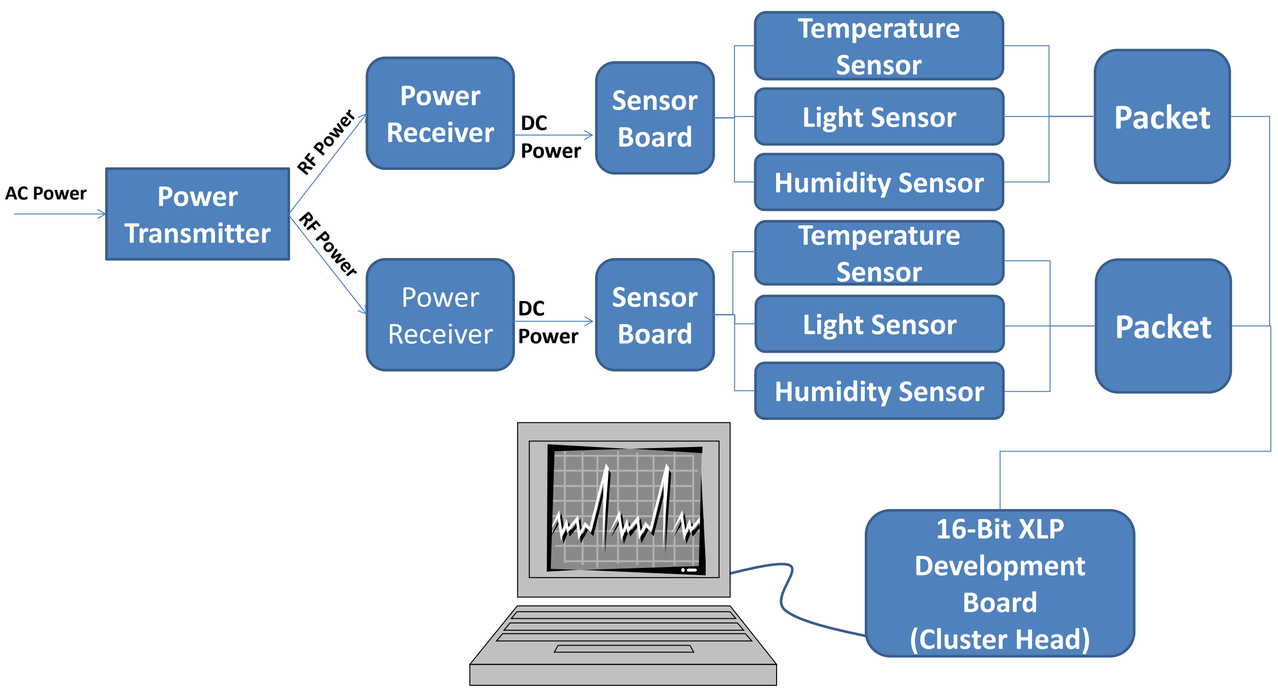
- Purchased and configured Powercast's P2110-EVAL-01 Development Kit.
- Developed software to visualize sensor data in real time.
- Created three algorithms, to help monitor the health of the network and calculate when a sensor node has failed.



Algorithms:

1. Short Term Analysis
2. Long Term Analysis
3. Zero Value Analysis

Wireless Sensor Network Flowchart

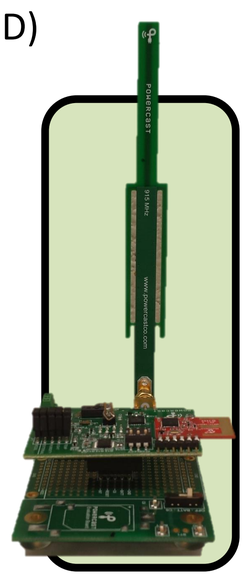


Budget

Product	Cost
Powercast P2110 EVAL-01 Development Kit	$1250
Total	$1250

Hardware

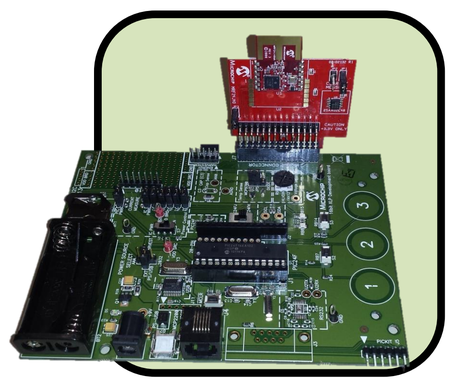
- Power and Data Transmitter (TX91501-3W-ID)
- (2) P2110 Evaluation Board (92110-EVB)
- (2)Patch Antennas
- (2)Dipole Antennas
- (2)Wireless Sensor Board (WSN-EVAL-01)
- Microchip 16-bit XLP Development Board (DM240311)
- Microchip MRF24J40 PICtail/PICtail Plus Daughter Board (AC164132-1)
- PICkit Programmer/Debugger (PG164130)



92110-EVB

TX91501-3W-ID

DM240311

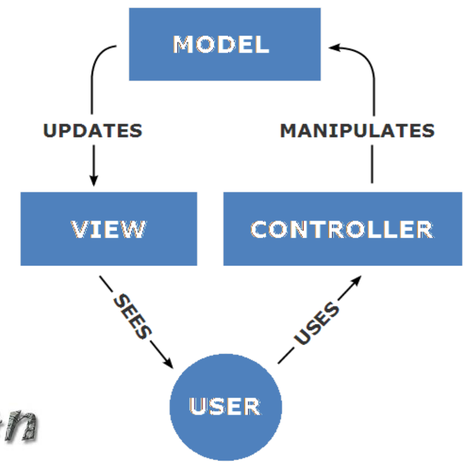


Software

Language:
Python 2.7

Python Libraries:

PySerial
matplotlib
wxPython



Architecture:
The software is organized in a model-view-controller architecture as shown above to the right. The incoming data bytes are read over a serial USB connection into the controller. The controller then checks the integrity of the data and passes it to the model for storage. The model periodically runs diagnostic tests to check for failing and failed nodes. The view displays the status of the network and graphs incoming sensor data in real time. An operator can also retroactively view any time range of data over the history of the network.