KNELL: Keen Nameless Electronic Loss Lookout

705-power-loss-detection

A cellular Internet of Things (IOT) device that can detect power loss from main AC. Designed as a system as a whole to send out notifications to facility managers/ Staff of interest so that memebers can perform graceful shutdown of key Information Technology infrastructure and services withon alloted Uninterrupted Power Supply (UPS) alloted time.

Particle software orginization

`/src` folder:

This is the source folder that contains the firmware files for your project. It should *not* be renamed. Anything that is in this folder when you compile your project will be sent to our compile service and compiled into a firmware binary for the Particle device that you have targeted.

If your application contains multiple files, they should all be included in the `src` folder. If your firmware depends on Particle libraries, those dependencies are specified in the `project.properties` file referenced below.

`.ino` file:

This file is the firmware that will run as the primary application on your Particle device. It contains a `setup()` and `loop()` function, and can be written in Wiring or C/C++. For more information about using the Particle firmware API to create firmware for your Particle device, refer to the Firmware Reference section of the Particle documentation.

`project.properties` file:

This is the file that specifies the name and version number of the libraries that your project depends on. Dependencies are added automatically to your `project.properties` file when you add a library to a project using the `particle library add` command in the CLI or add a library in the Desktop IDE.

Current libraries in use

DiagnosticsHelperRK : a library to access lower level things inside of the Device OS that particle boards run. Specifically, we use it to read the system power source. Credit to rickkas7: https://github.com/rickkas7/DiagnosticsHelperRK MIT license type so can be utilized for proprietery/ commercial use.

Projects with external libraries

TODO: we need to restructure the simple library use in this repo to match the below:

If your project includes a library that has not been registered in the Particle libraries system, you should create a new folder named `/lib/<libraryname>/src` under `/<project dir>` and add the `.h`, `.cpp` & `library.properties` files for your library there. Read the Firmware Libraries guide for more details on how to develop libraries.

Note that all contents of the `/lib` folder and subfolders will also be sent to the Cloud for compilation.

Device setup

Hardware required:

BORON 404x LTE (USA) Markup : picture alt

STOCK ANTENNA: Markup : picture alt

Battery (any 3.7 LIPO): Markup : picture alt

Follow the Boron 404x quickstart: https://tools.particle.io/setup Specifically, there may be some issues with the default firmware/ version of device OS that came installed on the device. Please visit the device doctor for diagnosing any problems encountered with the device not connecting to the cloud properly: https://docs.particle.io/tools/doctor/

Once firmware is flashed properly you should have success for connection to the particle cloud. It is recommended you download the mobile app onto your device and log onto the particle cloud console https://console.particle.io/ to see your device connected.

From here you can do any of the LED flash setups/ tests to ensure you are understanding the Particle object etc. part of this code **currently** maintains some test code and explanations on how to use the Particle objects/ cloud.

System description

The entire system is composed of a device layer, a cloud layer, a service layer, and a user layer. See the following diagram for a visual model: Markup : picture alt

Account credentials

TBD

Push notification service: Pushover

Why? it is $$$$cheap$$$$ TBD

Particle Console webhook setup

TBD WIP