DataGatheringSystem

Overview

Sample software system used to monitor data from various sensors.

Files

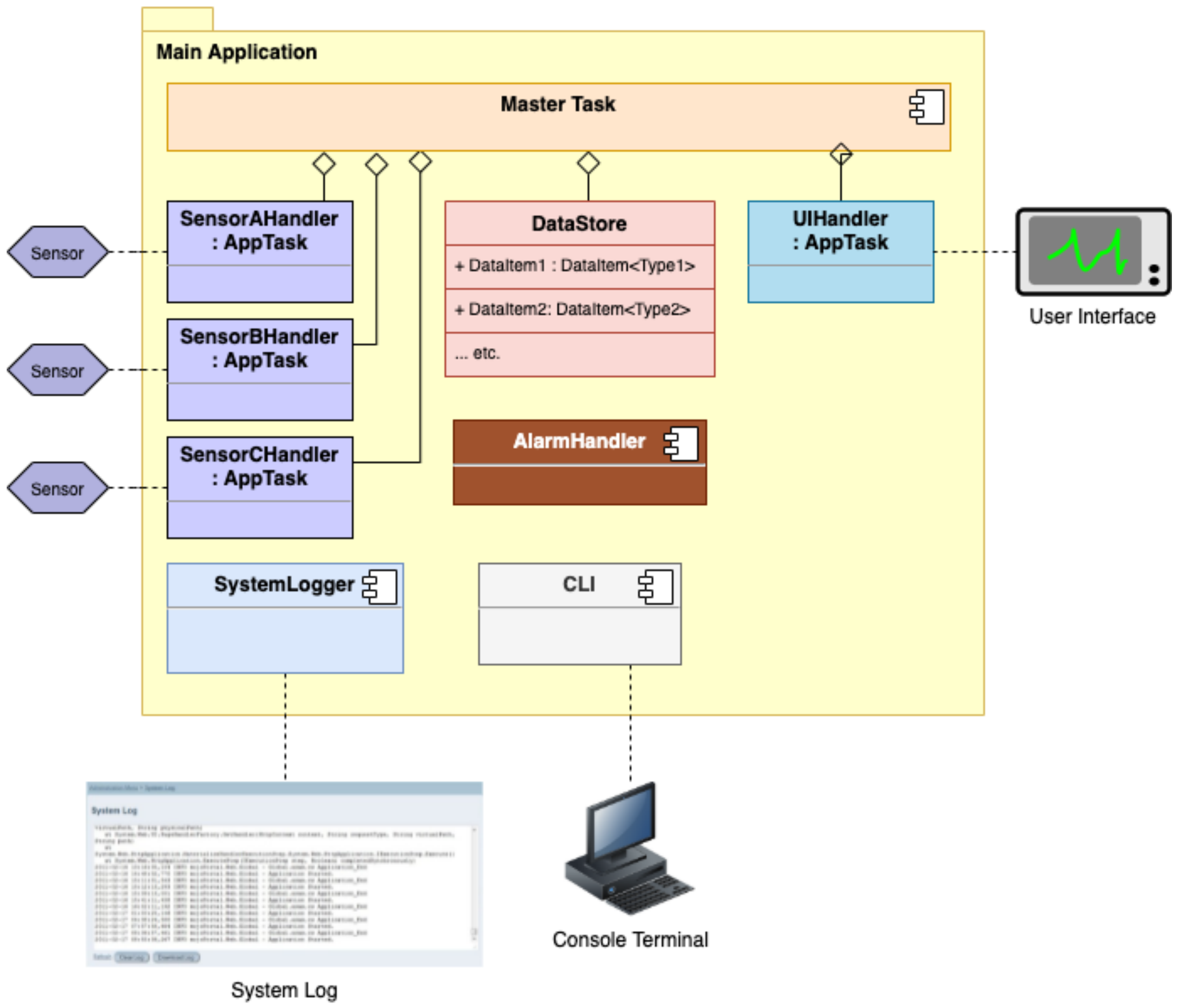
```
DataGatheringSystem
    ├── Makefile
    ├── DataMonitor.cpp
    ├── DataStore.cpp
    ├── DataStore.h
    ├── FluidTempSensor.cpp
    ├── FluidTempSensor.h
    ├── LICENSE
    ├── Makefile
    ├── MasterTask.cpp
    ├── MasterTask.h
    ├── README.md
    ├── SensorTask.cpp
    ├── SensorTask.h
    ├── UITask.cpp
    └── UITask.h
```

External Dependencies

- CPP Tools library
  Provides the implementation for AppTask, Command Line Argument parsing, Data Items, time stamps, and the System Logger.

Description

Data Gathering System Software Architecture



The main() function of the application provides the "background" processing for the application. All data gathering, handling, and logging is done using subordinate application tasks, which are controlled by a master task (which in turn can be controlled by the main function.)

The data to be monitored is owned by an overall data-store object. Each data item in the data-store is a publish-subscribe object that maintains its own value, state (valid, out-of-range, stale), ID, units, upper and lower value limits, and time-to-expiration (staleness).

Each hardware sensor is accessed and controlled by a subtask, which is instantiated by the main application, given parameters it needs to identify itself, the data item it's updating, the hardware sample rate, and the reporting time (used for averaging samples).

Any task can subscribe to a data item to be notified of a change. All data items can also be polled any time to get its value and state. The UI Task polls data items on a regular basis to update the user interface. Since each data item contains its own name, units, value, and state, this makes it easier to display the item in a generic way.

The CLI (TBD/TODO) can be used for maintenance, configuration, and/or control of the system.

An Alarm Handler (TBD/TODO) can be used for reporting/logging system error conditions, as well as invalid, expired, or out-of-range data.

The System Log contains as much or as little system information as desired, depending on the log level the system is configured for (Critical, High, Medium, Low, Info, Debug). By default, the system would normally be set to allow as low as Low or Info levels. For develpment, the Debug level is recommended.
Multiple loggers can be used for various reasons, such as to maintain a separate log file for each type of data, or for multiple parts of the system.