DataFlow OpenDNP3 Outstation API Functions

Open-Source Libraries:

- OpenDNP3 v3.0.4 – no modifications made to OpenDNP3 codebase
- RapidJSON v1.1.0 – no modifications made to RapidJSON codebase

Data Flow Outstation API:

- Source included in code.zip and also in:
  - …/opendnp3-dataflow/cpp/lib/src/api/api.cpp
- Compiled object in objects.zip and also in:
  - …/opendnp3-dataflow/build/cpp/lib/CMakeFiles/opendnp3.dir/src/api/api.cpp.o

DataFlow OpenDNP3 Outstation API Functions

Start Oustation thread with 150 database indices for each type:

- int UseAsOutstation(int port, std::string OutstationIP, int MasterAddress, int OutstationAddress, bool unsolicitedEnabled)

  - Port = DNP Port (ie. typically 20000)
  - OutstationIP = IP Address of the Outstation (ie. 192.168.x.x)
  - MasterAddress = DNP Master Address (ie. 1)
  - OutstationAddress = DNP Outstation Address (ie. 10)
  - unsolicitedEnabled = TRUE for enabled

  This will enable the Outstation and start communications with the Master and then launch a loop to read analogs, digitals, controls, setpoints, counters and freeze.

Update Database Values:

- bool queueDigitalEvent(int index, bool value, bool hasTime, DNPTime time)
  - Index = OpenDNP3 Outstation Database Index
  - Value = Boolean value to write to the BinaryStatus
  - hasTime = flag to determine if time is being passed to the event
  - time = DNPTime (not time_t) of the event. Struct DNPTime used by OpenDNP3 is included in the api.cpp for reference in order to determine how to send time from HSS.

- bool queueAnalogEvent(int index, double value, bool hasTime, DNPTime time)
    - Index = OpenDNP3 Outstation Database Index
    - Value = Double value to write to the AnalogStatus (OpenDNP3 uses doubles for analogs)
    - hasTime = flag to determine if time is being passed to the event
    - time = DNPTime (not time_t) of the event.   Struct DNPTime used by OpenDNP3 is included in the api.cpp for reference in order to determine how to send time from HSS.

- bool controlActionDigital(int index, bool value, bool hasTime, DNPTime time)
    - Index = OpenDNP3 Outstation Database Index
    - Value = Boolean value of the BinaryOutputStatus. This is the current state of the control, not the control action itself as that is executed in the OpenDNP3 library.
    - hasTime = flag to determine if time is being passed to the event
    - time = DNPTime (not time_t) of the event.   Struct DNPTime used by OpenDNP3 is included in the api.cpp for reference in order to determine how to send time from HSS.
    - This will then Write a JSON object to /tmp/outstation.json with the following members: {"**control**", value, index} for use by the TCU to execute

- bool controlActionDigital(int index, double value, bool hasTime, DNPTime time)
    - Index = OpenDNP3 Outstation Database Index
    - Value = Double value of the AnalogOutputStatus. This is the current state of the setpoint, not the setpoint action itself as that is executed in the OpenDNP3 library.
    - hasTime = flag to determine if time is being passed to the event
    - time = DNPTime (not time_t) of the event.   Struct DNPTime used by OpenDNP3 is included in the api.cpp for reference in order to determine how to send time from HSS.
    - This will then Write a JSON object to /tmp/outstation.json with the following members: {"**setpoint**", value, index} for use by the TCU to execute

- bool queueCounterEvent(int index, uint32_t value)
    - Index = OpenDNP3 Outstation Database Index
    - Value = Integer value to write to the Counter (non-negative!)

- bool freezeCounterEvent(int index, bool value, UpdateBuilder& builder)
    - Index = OpenDNP3 Outstation Database Index
    - Value = Boolean value to execute the Counter Freeze (ie. TRUE, FALSE)