EPICS Control of Wireless Sensors

BROOKHAVEN NATIONAL LABORATORY

Michael Rolland, Department of Computer Science, Stony Brook University, Stony Brook, NY 11794
Kazimierz Gofron, Department of Photon Sciences, Brookhaven National Laboratory, Upton, NY 11973

NATIONAL SYNCHROTRON LIGHT SOURCE II

Abstract

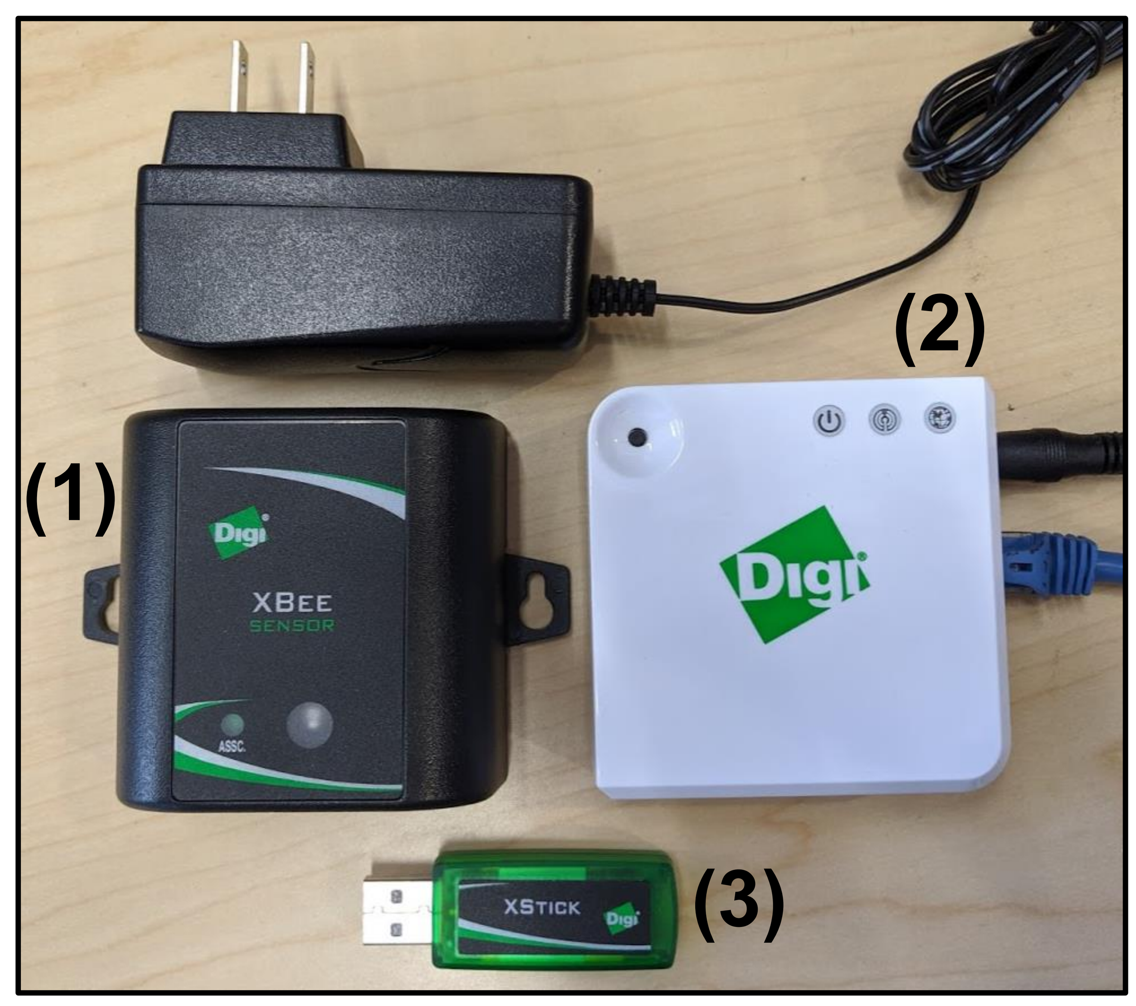
At the trade-off of power, wireless technologies are much more portable and convenient than their wired counterparts. This is especially true in the scientific sphere, where many environmental factors must be recorded at all times at as many locations as possible. Using these technologies, scientists can often reduce cost while maximizing the number of sensors without compromising sensor quality. To this end, we have developed EPICS controllers for both Bluetooth Low Energy (BLE) sensors and Zigbee sensors. For BLE, we chose the Nordic Thingy:52 for its low cost, high battery life, and impressive range of sensors. The controllers we developed combine EPICS base functions, the Bluetooth generic attribute data structure library, and multithreading techniques to enable real-time broadcast of the Thingy's 20+ sensors' live values. Because BLE is limited in range, we also developed a controller for an XBee sensor which, through the Zigbee mesh protocol, can expand its range through each node added into the network. With these controllers, NSLS-II scientists will have access to a whole new class of sensors which are both easier to deploy and cheaper than their wired predecessors.

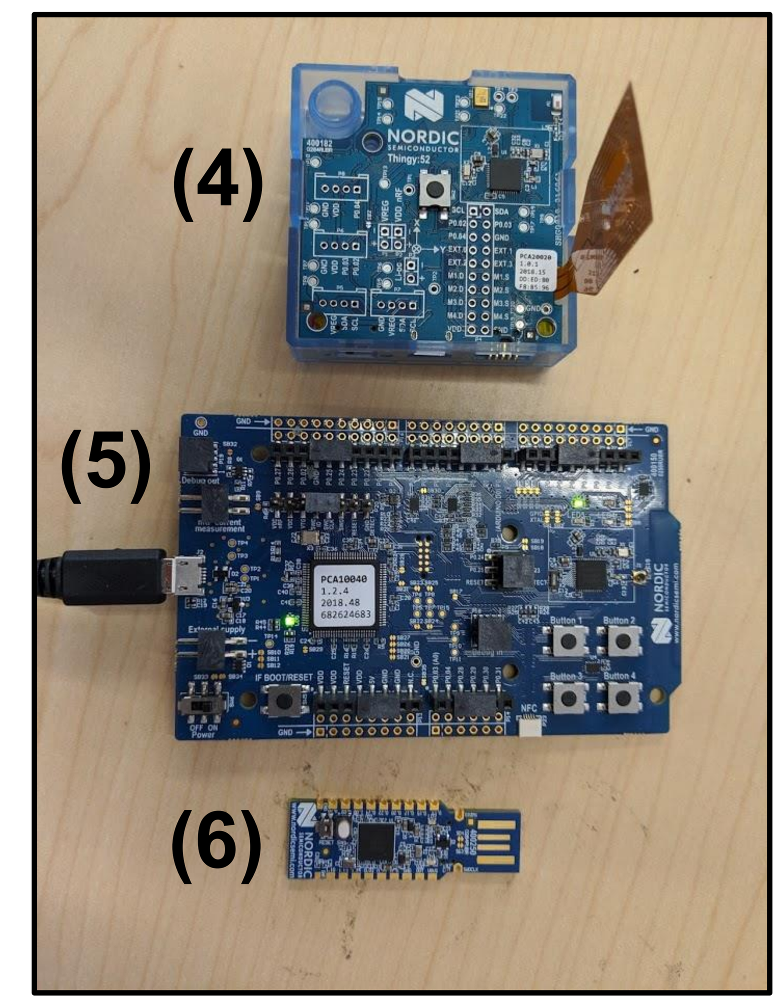
Sensors

We first explored wireless sensors with the Digi XBee L/T/H sensor **(1)**, which uses the ZigBee protocol and includes sensors for light, temperature and humidity for $100. For ZigBee connectivity we used both the ZigBee Gateway **(2)** and XBee XStick **(3)**. For Bluetooth we chose the Nordic Thingy:52 **(4)**, which has an impressive array of environment & motion sensors at $40. For developing custom firmware we used Nordic's NRF52 DK **(5)** and for Bluetooth connectivity we used Nordic's NRF52840 dongle **(6)**. We developed two types of custom firmware for the Thingy:52 and NRF52 DK:

Bluetooth Mesh
- Many-to-many connection
- One 'bridge' Thingy:52, up to 9 'node' Thingy:52s
- Requires radio to be on constantly; battery drain

Bluetooth Multi-Link
- Many-to-one connection
- Central NRF52 DK connects to up to 19 Thingy:52s
- Low energy connection between nodes and central





The IOCs

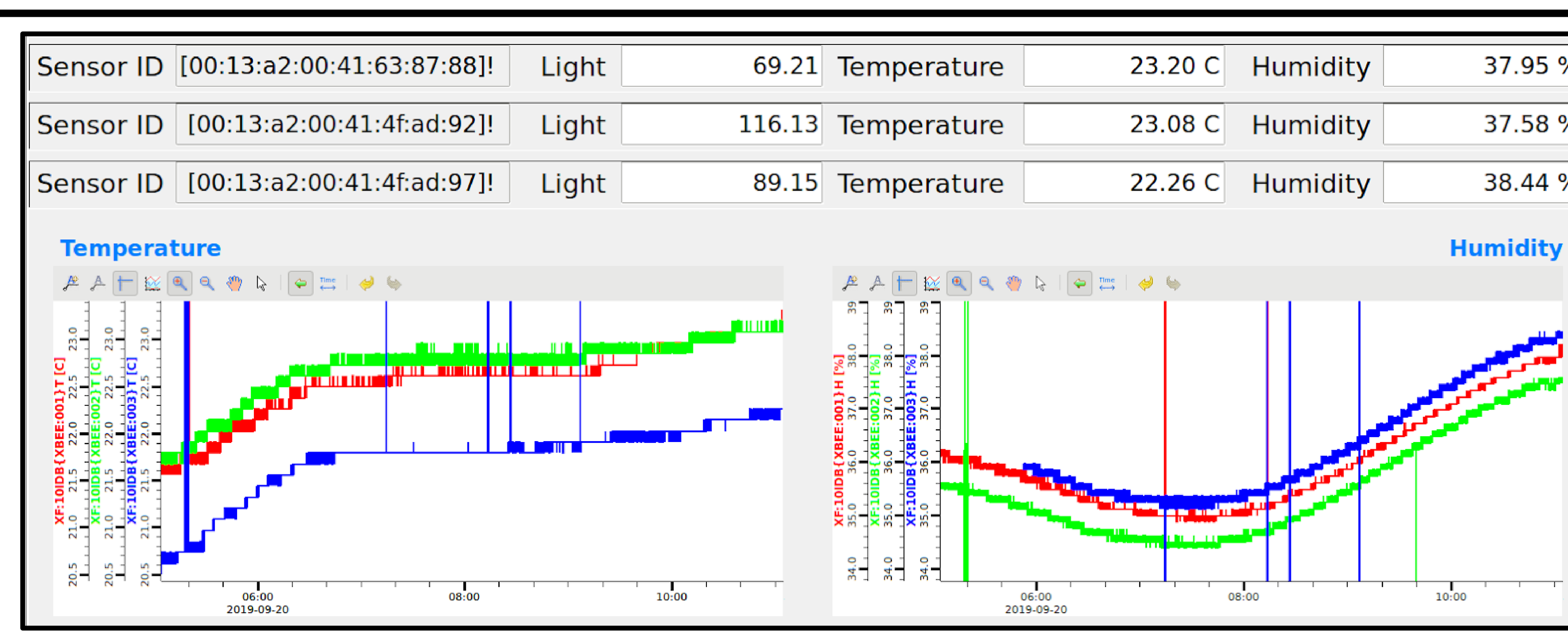
(1) XBeeIOC
- Connects to XBee L/T/H sensors using either an XBee Gateway or XBee XStick
- Uses StreamDevice structure, interfacing with independent Python script running a simple TCP server
- Finicky, prone to failure

(2) ThingyIOC
- Connects to one Thingy:52 with Bluetooth low energy
- Supports all sensors
- Not scalable; 1 USB slot for 1 sensor

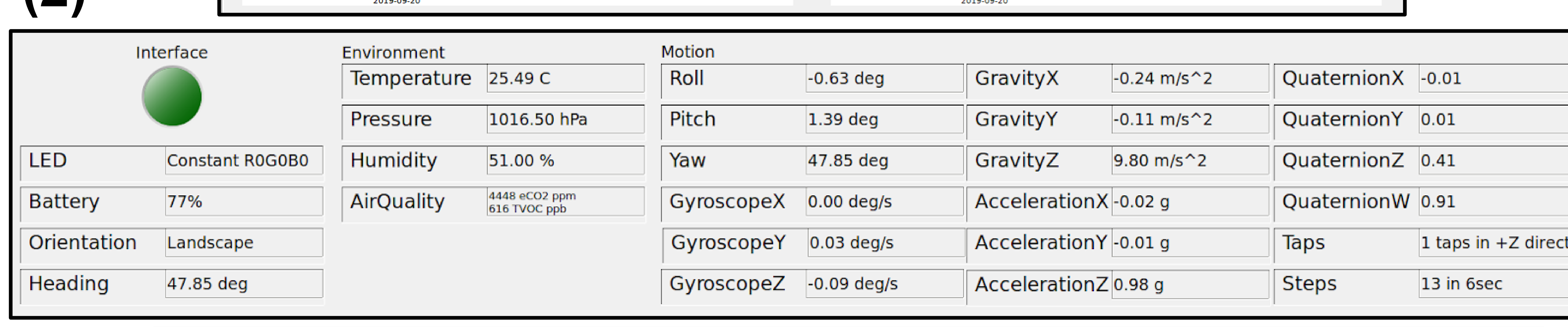
ThingyMeshIOC
- Connects to bridge Thingy:52 which connects to up to 9 node Thingy:52s
- Supports temperature, pressure, humidity
- Poor battery life; several days

(3) ThingyAggregatorIOC
- Connects to NRF52 DK which connects to up to 19 node Thingy:52s
- Supports all sensors
- Loss of approx. 1% in 24 hours

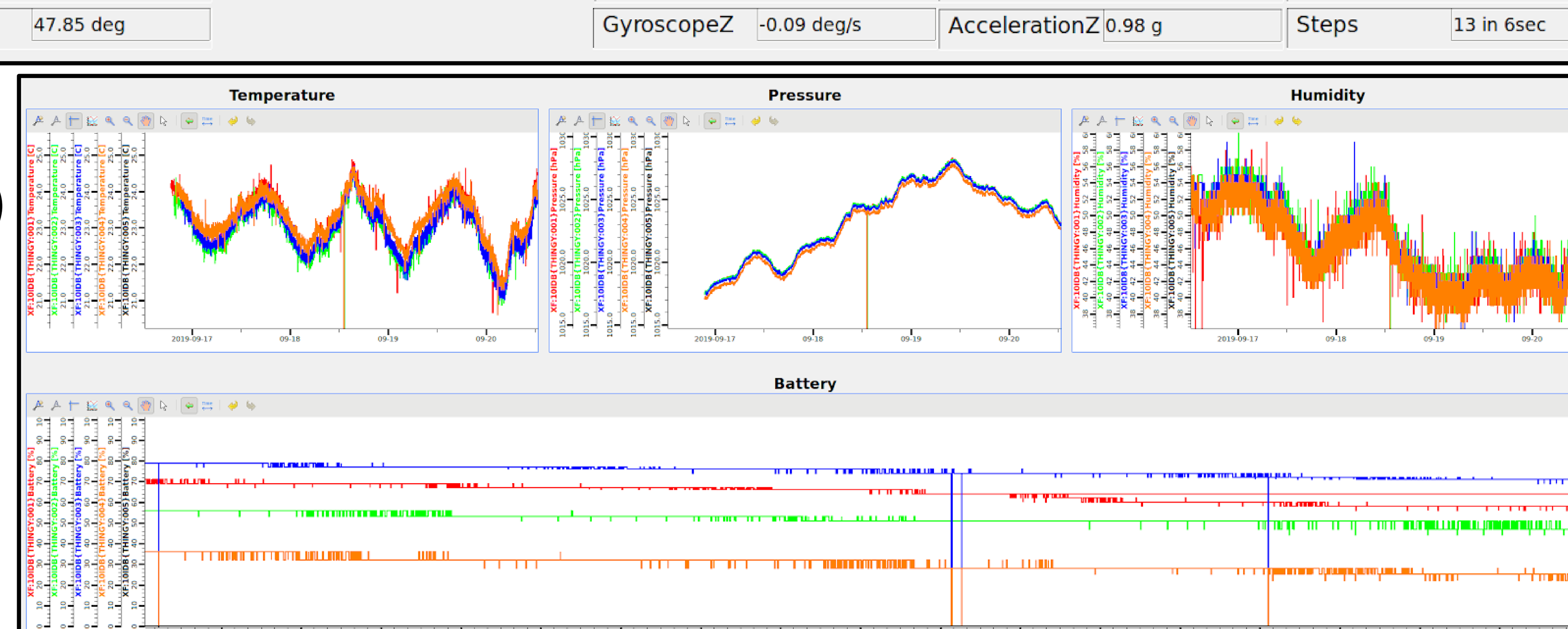
(1)



(2)



(3)



Going Forward

These IOCs can be run on the extremely compact Raspberry Pi 3 equipped with a Nordic Bluetooth dongle, and thus deployment to the beamlines will be simple and painless. With the multi-link network, one USB dongle can read from up to 19 sensors which last roughly 3 months with a full charge. Next steps include fixing any performance bugs on the IOCs and identifying a sensor capable of Bluetooth mesh networking without extreme battery loss so we can capitalize on the extensive range of Bluetooth mesh.

Stony Brook University College of Engineering and Applied Sciences

U.S. DEPARTMENT OF ENERGY Office of Science