Telemetry basics

- Our elab is monitored and controlled using a language called xml, or extensible markup language.
- You recall html stood for hypertext markup language, well this is another "markup" language that holds data not text.
- You have also see this in your word documents (.docx) spreadsheets (.xlsx) and slide shows (.pptx), where "x" stands for xml, or a container for data.
-

Part 1: Reading xml data

- Scan the elab network, looking for devices in the format 10.14.x.1
- On each webpage you will see data presented
- In the browser window, enter the ip address then /state.xml to see the data inside each container
- Telemetry is the reading of remote data.

Part 2: Controlling devices with xml

Control can also be done using xml in the following format:

Output State Control: Command Description

- state.xml?relay1State=0 Turn Output 1 *OFF*
- state.xml?relay1State=1 Turn Output 1 *ON*
- state.xml?relay2State=0 Turn Output 2 *OFF*
- state.xml?relay2State=1 Turn Output 2 *ON*

Pulse Relay: Command Description

- state.xml?relay1State=2 Pulse Output 1 for the preset time (1.5 seconds).
- state.xml?relay1State=2&pulseTime1=5 Pulse Output 1 for 5 seconds.
- state.xml?relay1State=2&pulseTime1=5 Pulse Output 1 for 5 seconds.
- state.xml?relay1State=2&pulseTime1=5 Pulse Output 1 for 5 seconds.
- state.xml?relay1State=2 Pulse Output 1 for the preset time (1.5 seconds).

You will setup a device for you to control, please only control devices with my permission, as some are mission critical to the elab.

Turning on lights and opening windows are fine, but nothing else for now

Try reading devices around the elab with the http://ipaddress/state.xml web link

Try controlling these devices using the commands above:

- 10.14.17.1
- 10.14.18.1
- 10.14.19.1

Part 3: sensors
We will begin with measuring temperature.
Two types of temperature sensors are most common:
2 wire sensor: also called a thermistor is a resistor that changes value (resistance) when temp changes. You will see these in some solar thermal controllers, or older pre-digital devices.
3 wire digital sensor: these have three wires: power (+), ground (-) and data. Each sensor has a digital address. These are much more accurate than 2 wire sensors, but require more advanced circuits to use

Temp sensor setup:

1. get a control by web temperature module, power supply (12V) and a digital temp sensor (metal case)
2. wire the power and sensor to the device, check to see if the power light is on (see below)
3. on a desktop computer, create a network connection for the 192.168.1.0/24 network
4. open a web page for 192.168.1.2, the address of CBW devices out of the box
5. go to 192.168.1.2/setup.html with the login admin/webrelay and change the IP address to the one given by me
6. restart the unit
7. go to the new address, and go to the temperature page

8. add the digital address for the three wire sensor
9. check the web page to see if it records the correct value (warm with your hands to see the change)
10. use xml to check the values: http://ipaddress/state.xml
11. use xml to control the relays in the unit (see above for commands)
12. verify that you can control the unit by watching the web page change
13. expand your system by adding more digital sensors (note they all connect to the same terminals)

Temperature module manual:

Download temperature_um_v1.6.pdf (application/pdf, 4.6 MB)

Advanced next steps:

- Analog module: measure current and power
- X-320 module
- X-420 module
- 400 series modules (bus controlled)
- arduino temp sensors
- raspberry pi sensors
- Lorawan sensors (fire detection project)

Lorawan sensors: Drone project and others...
Lorawan stands for long range, wide area network.
These units are very low power, very small data systems used for remote telemetry (data sensors and control).
Advantages:

1. very low power, some run on batteries for several years, so no need to wire power supplies
2. radio linked using 900 mHz radio, so no need for network wiring
3. long range: can be used up to several miles away, depending on terrain
4. flexibility: move them around as you like, no need to reconfigure
5. location tracking: some have built in GPS

We will be testing two types of sensors: SenseCap sensors and Dragino sensors.
SenseCap sensors connect to a dashboard provided by the company:

https://sensecap.seeed.cc/portal/#/dashboard
They can also be connected to non-sponsored systems like The Things Network (TTN) or Thingsboard
Dragino sensors are a bit cheaper, and can only be integrated into either TTN, Thingsboard or your own MQTT database server.
MQTT is a database protocol used by small devices, holding and transferring either messages (like in Meshtastic) or data.
We use tools like MQTT explorer to check the data going in and out:

https://mqtt-explorer.com/

Temperature data is not usually encrypted (who cares?), but Meshtastic has an encryption option, so messages can be encrypted.
Fun fact: if you are a licensed radio operator, you can use higher power on Meshtastic, but you cannot use encryption, per the laws for radio operators.

Part 1: SenseCap

1. Login to the sensecap dashboard above and look at the data. You should see a weather station (3.27.25) and several sensors labeled alpha, beta and so on.
2. Check out the graphs available
3. Note the sensors, and possible data on location
4. Look at caltopo.com, this is were we need to add the location of the sensors

Part 2: The Things Network (TTN)

1. Login to TTN here: https://nam1.cloud.thethings.network/console/
2. Look for sensors and gateways
3. Look at traffic on the gateway, and the data stream
4. Adding a sensor is not easy, we will do this next

Part 3: MQTT

1. Check out the public mqtt server here: https://www.mqtt-dashboard.com/
2. Note the traffic
3. Using the mqtt explorer, see if you can watch data
4. Try watching Meshtastic messages using the server mqtt.meshtastic.org login meshdev/large4cats

Part 4: Mapping

1. Login to caltopo here: https://caltopo.com/p/0NE7J
2. Notice that the map can be shared.
3. Our goal is to create a single page dashboard for all of our sensors

---

Migrated from Apple Server Wiki. Original page ID: T8D7o8

Revision #3
Created 2026-09-29 15:59:30 UTC by Admin
Updated 2026-09-29 16:48:44 UTC by Admin