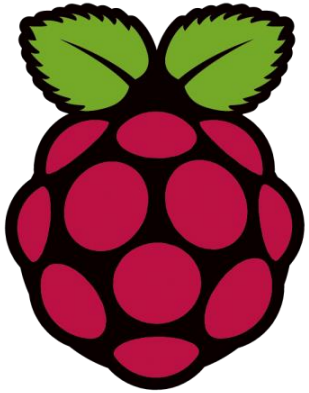


Indoormonitoring

LabVIEW on a Raspberry Pi and LabVIEW NXG Dashboard

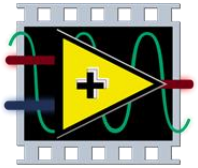
By

Wim Tormans

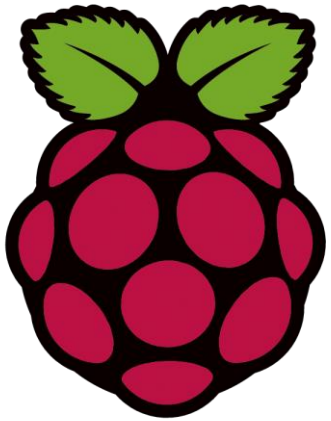
Wim Tormans

- Live in Belgium, work in the Netherlands
- Age: 38 / Married
- Father of 2 (Willeke 9y, Giel 7y)
- Hobbies: Photography, Running, LabVIEW
- Project Architect @ VI Technologies
- Using LabVIEW since 2008
- Using TestStand since 2013
- Certifications:

VI Technologies



LabVIEW



LabVIEW™NXG

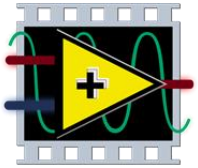
Agenda

- Why did I start this project
- What is the setup
- How do we program LabVIEW on a Raspberry Pi
- How to interact with your application on the Raspberry Pi
- Build a LabVIEW NXG Webvi
- Host the Webvi on a SystemLink server

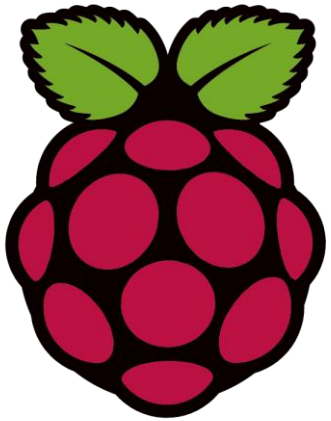
VIT echnologies

Why Did I start this project

- Thermostat in the wrong room
- Thermostatic cranes not broken
- Steffan = always cold
- Bas = always warm
- Jeffrey = always ‘configuring’ the thermostat and/or cranes



LabVIEW



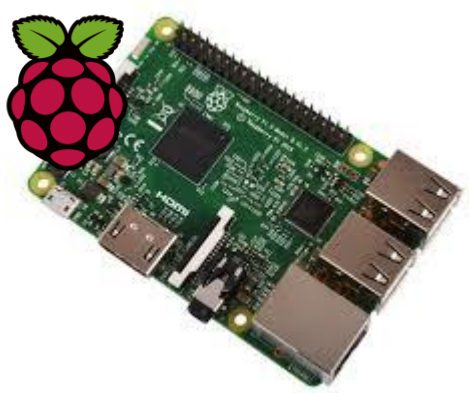
LabVIEW NXG

Agenda

- Why did I start this project
- What is the setup
- How do we program LabVIEW on a Raspberry Pi
- How to interact with your application on the Raspberry Pi
- Build a LabVIEW NXG Webvi
- Host the Webvi on a SystemLink server

VIT echnologies

What is the setup?



LabVIEW

LabVIEW NXG

- Raspberry Pi(3B) with I2C sensor
- Good old LabVIEW (2014)
- New LabVIEW NXG
- NXG Webmodule
- SystemLink Server

NXG Web Module

How do we program a RPI with LabVIEW

A few options:

- TS Experts for Raspberry Pi you can make UI's but has a license cost. Compiles to Python. Limited LabVIEW functions available.
- LabVIEW Makerhub LINX Library: you can't make UI's. It is FREE. All Functions.

I've chosen for the LabVIEW Makerhub:

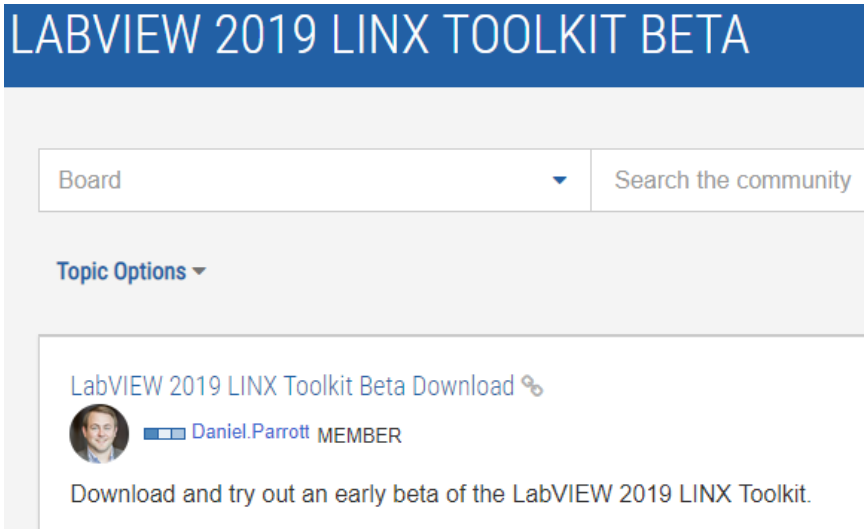
✓ Free

✓Easy to install

✓Support forum (a bit quiet lately)

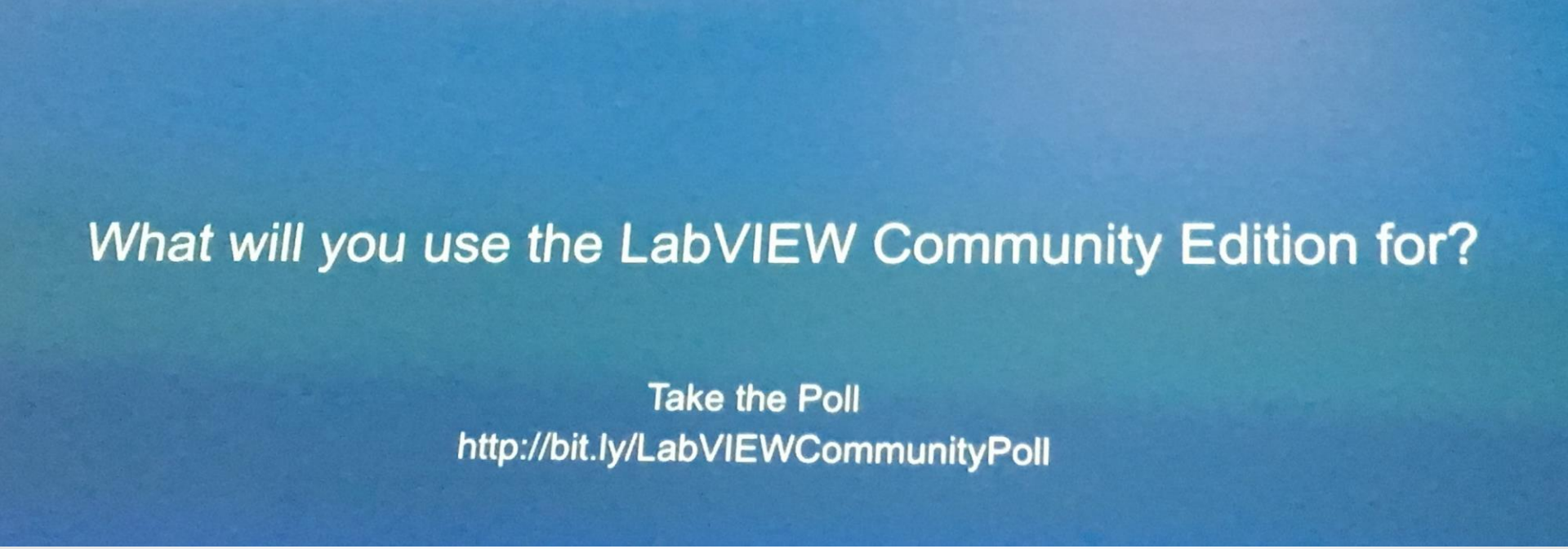
- LabVIEW 2014 but …

VI Technologies

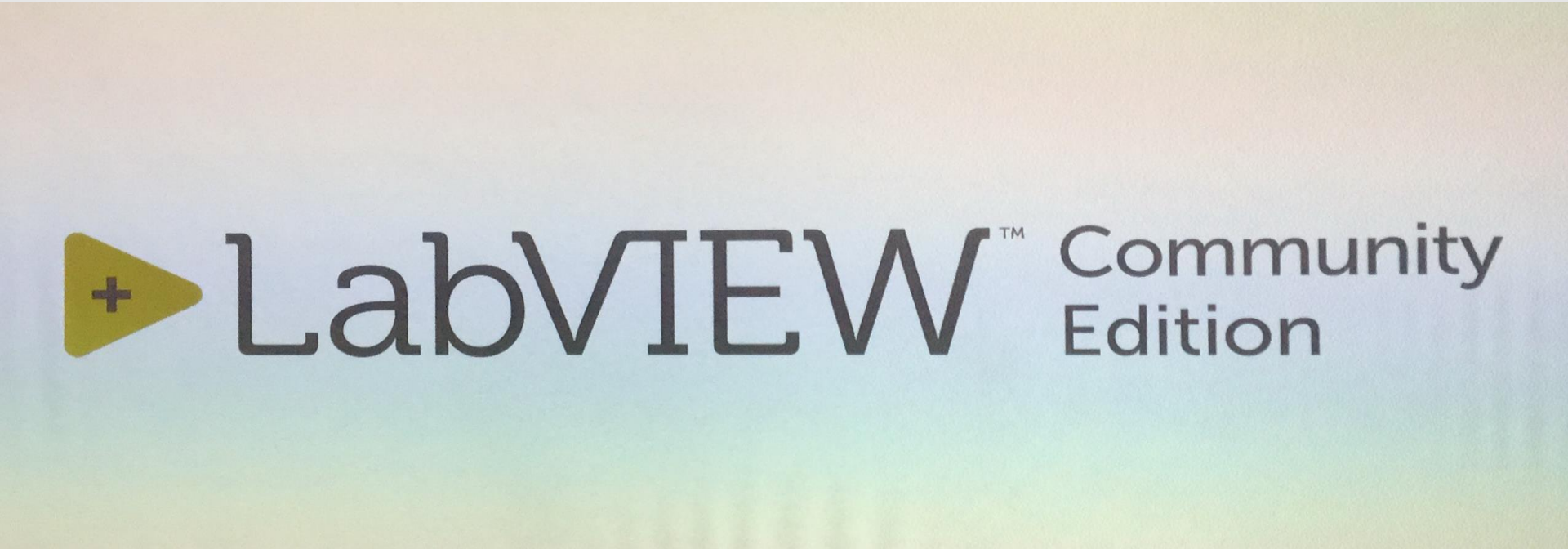


Technology Preview Program
Tested this morning → Only LabVIEW 2014 … for now…

Tweet by Fabiola De La Cueva @ GDevCon2



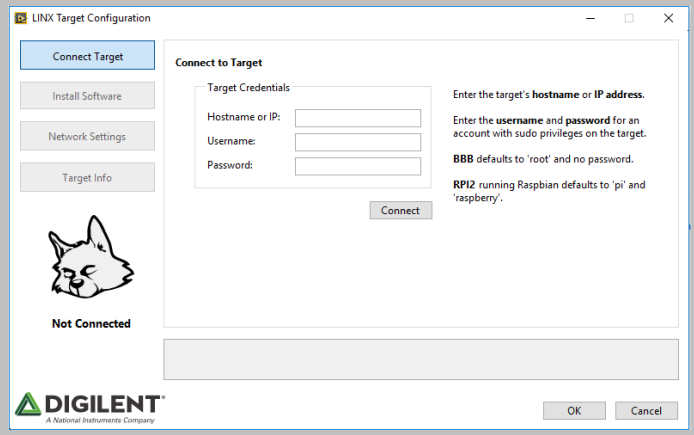
http://bit.ly/LabVIEWCommunityPoll

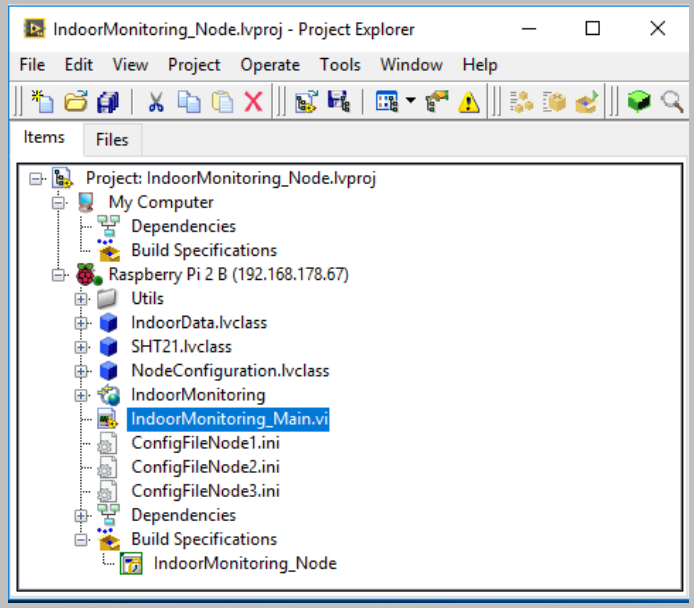


Raspberry Pi Installation “makerhub”

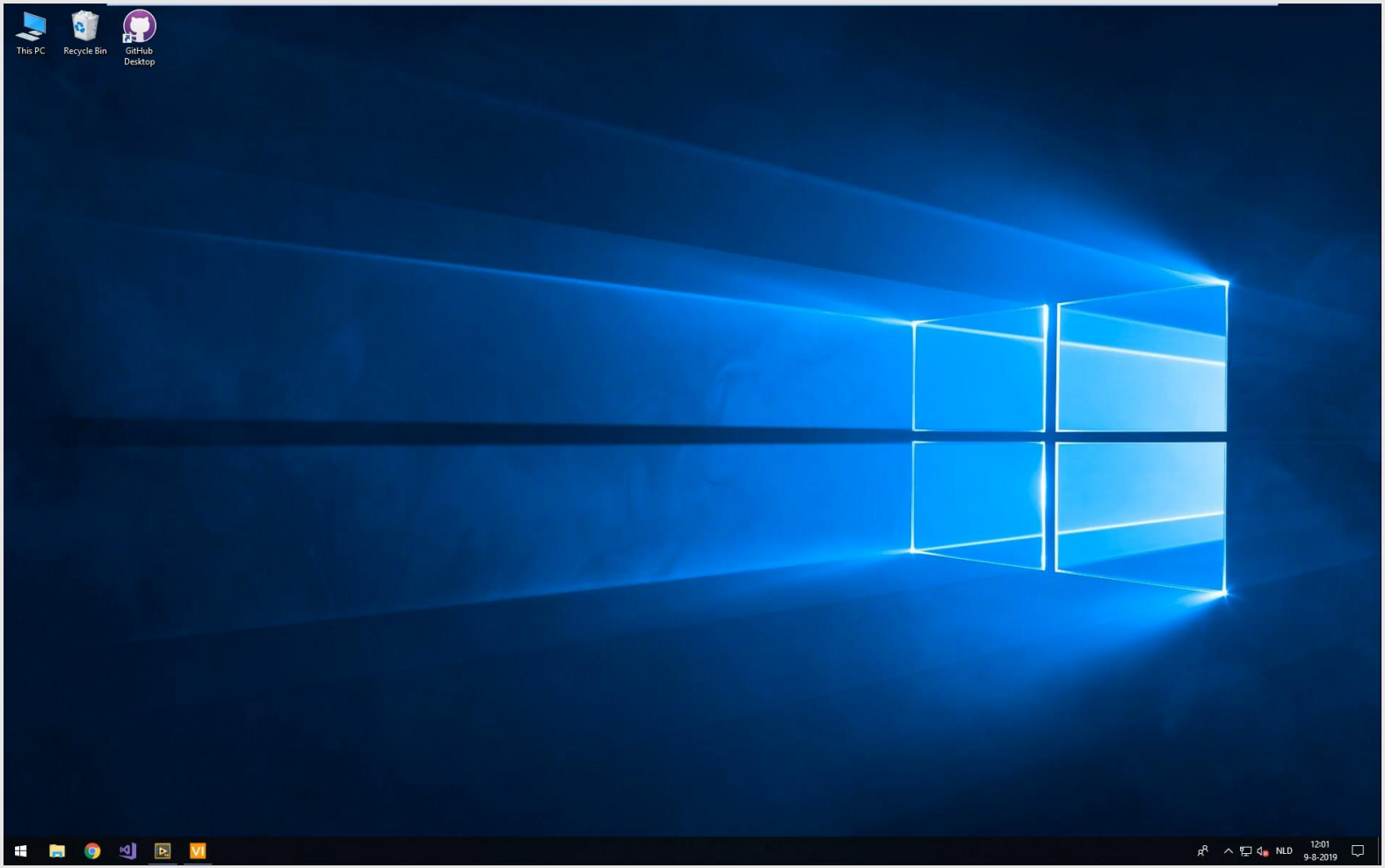
- Basic steps: (Tutorials are online with videos)
  - Download Raspbian image from raspberrypi.org
  - Flash the image with Win32DiskImager on uSD
  - Boot and Configure your RPI
  - Install LabVIEW 2014 + LINX library (VIPM)
  - Use the Target Configuration Wizard
  - Add a new LINX device to a project

- Extra Steps: (For RPI 3B):
  - SO linking fix
  - Optional fix for system CMDs



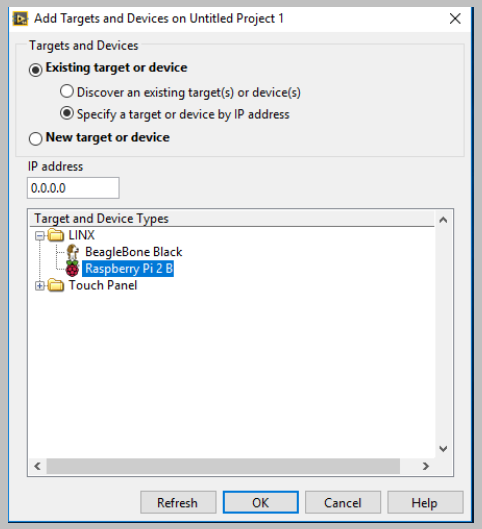


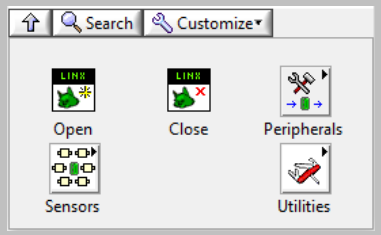
Installation on target demo

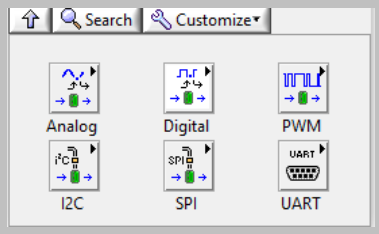


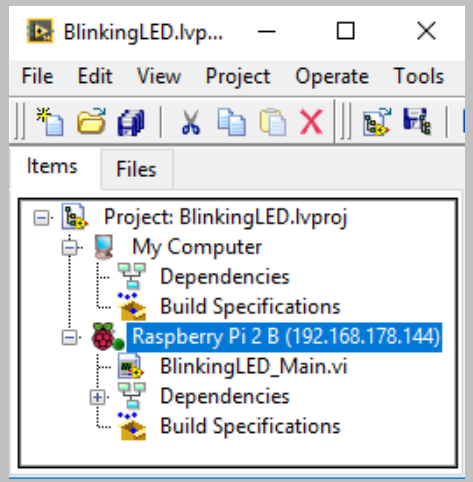
Raspberry Pi Programming

- Raspberry Pi target in project
  - Add target > LINX device > Raspberry Pi
- All standard LabVIEW functions are available
- Classes are supported (Also OpenGDS)
- LINX library to control IO



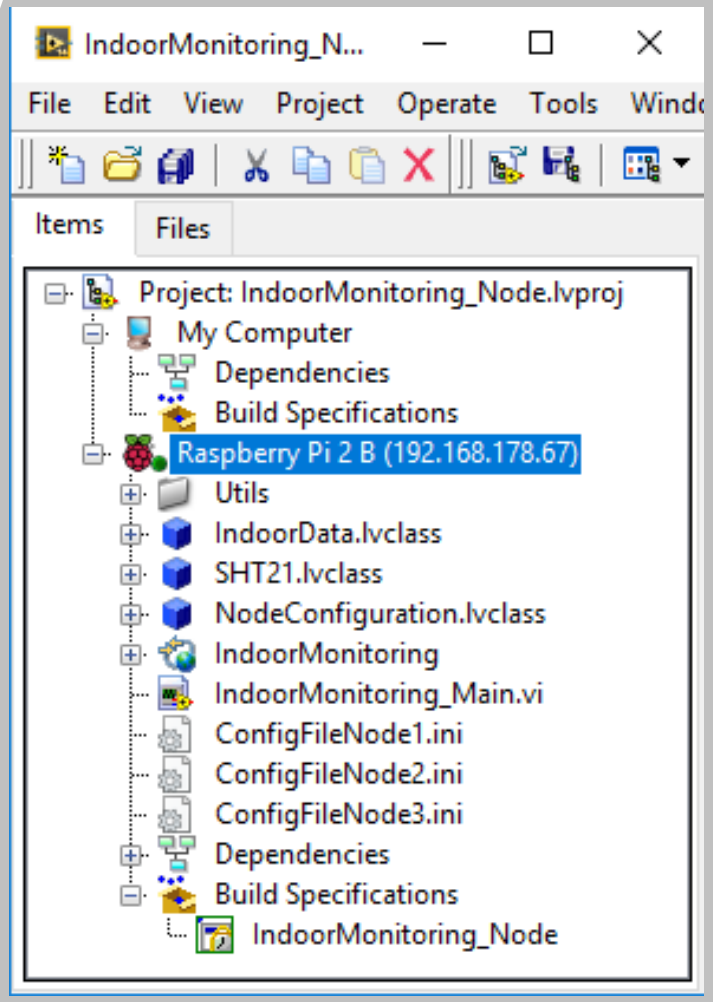






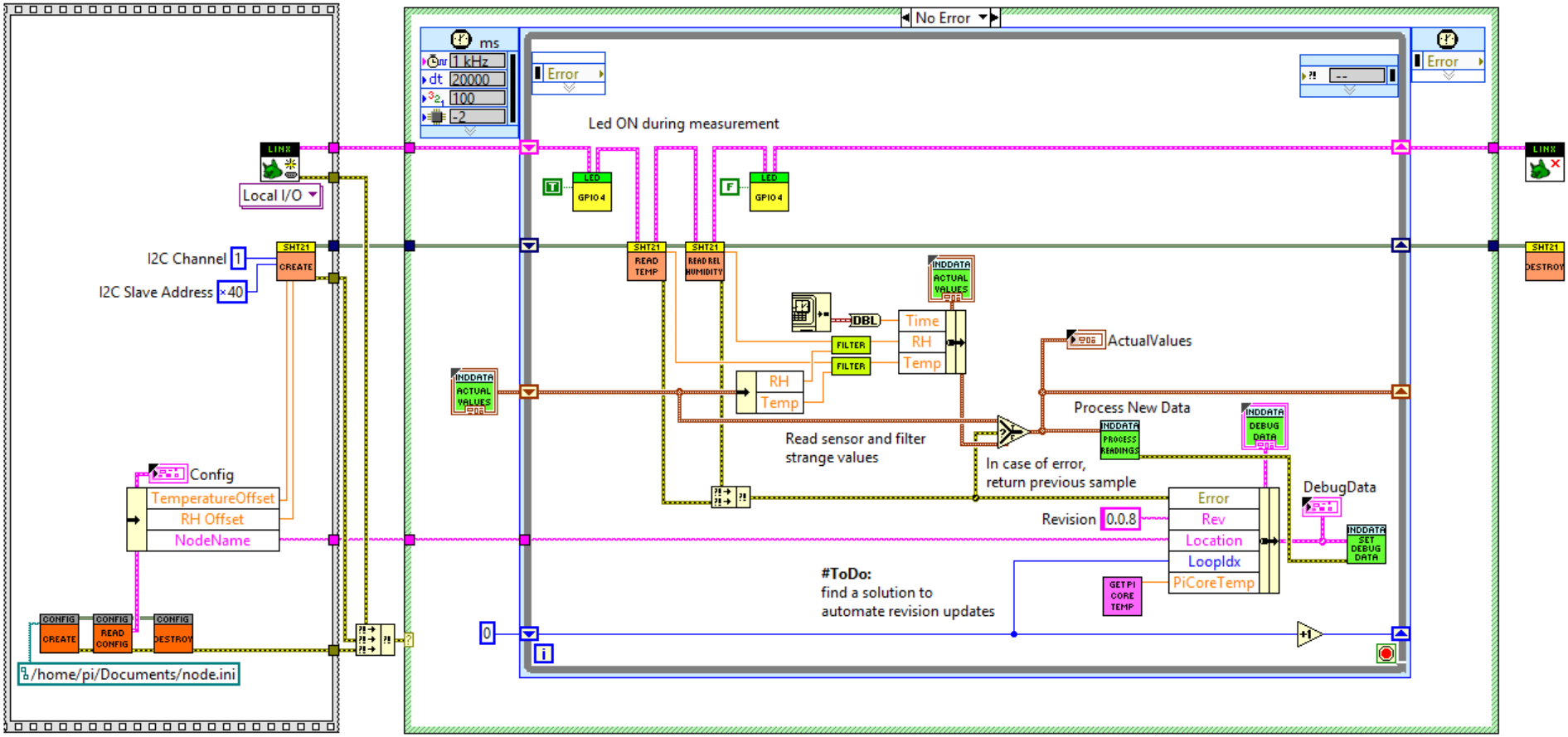
Raspberry Pi IndoorMonitoring

- IndoorData.lvclass (GOOP4 Singleton)
  - Processing of the measurement data
  - Used in main.vi AND in the webservice
- SHT21.lvclass
  - Sensor reading
  - Uses LINX library
- NodeConfiguration.lvclass
  - Configuration file
- IndoorMonitoring Webservice
  - API to access the data
- IndoorMonitoring_Main.vi



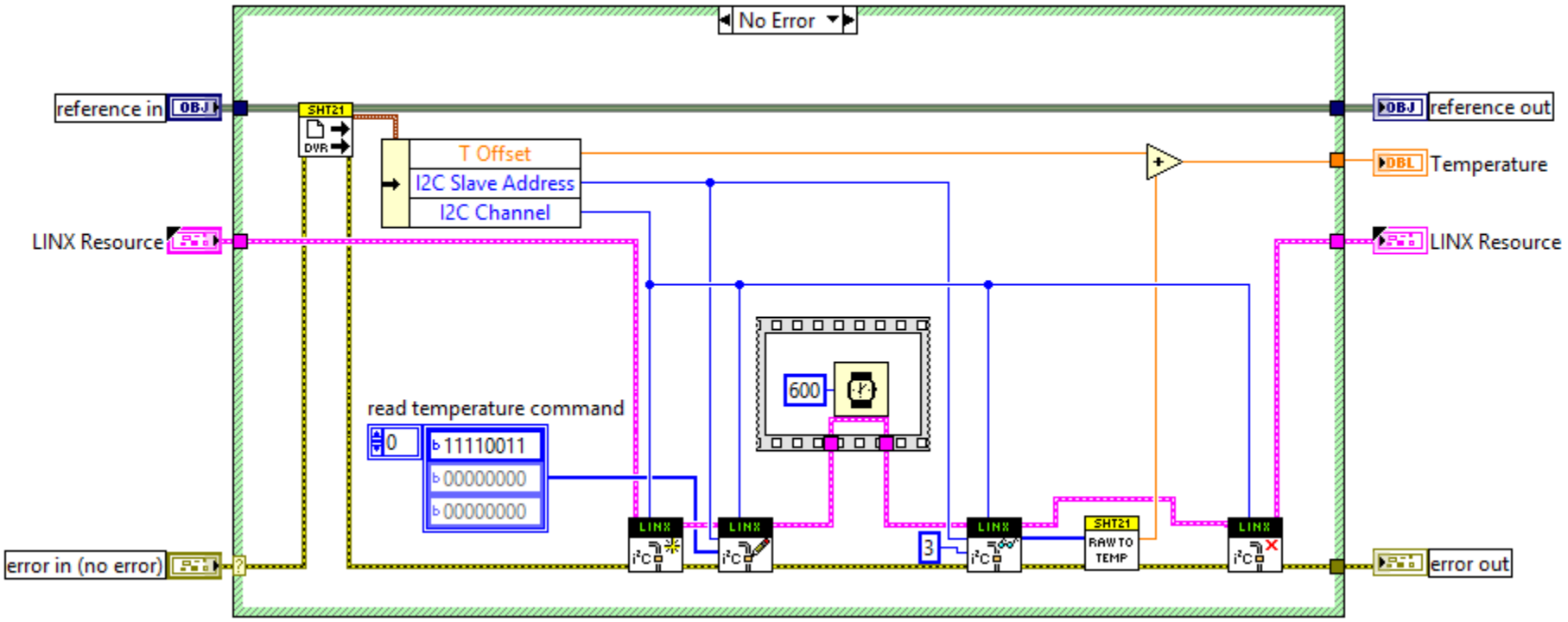
Raspberry Pi IndoorMonitoring

Main VI



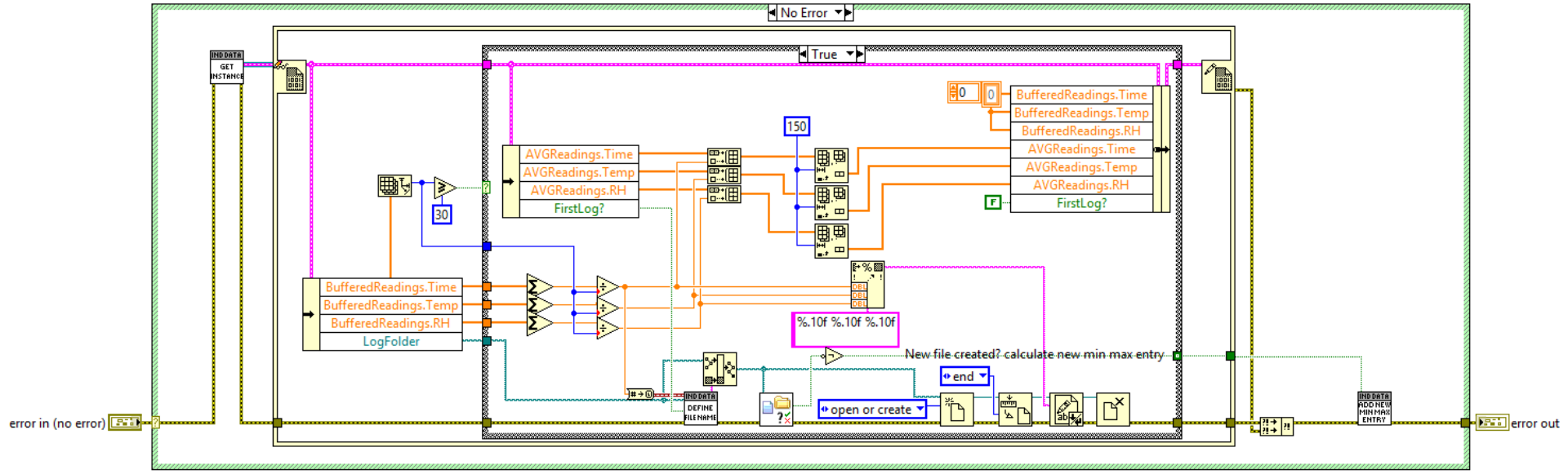
Raspberry Pi IndoorMonitoring

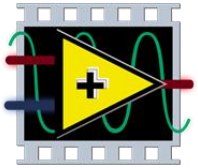
Read Temperature



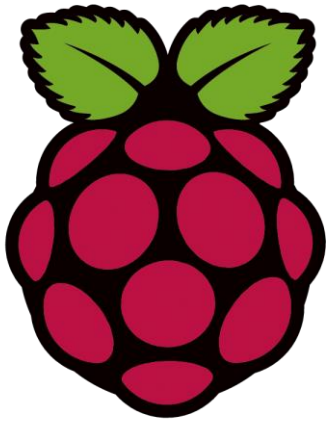
Raspberry Pi IndoorMonitoring

Process Data





LabVIEW



LabVIEW NXG

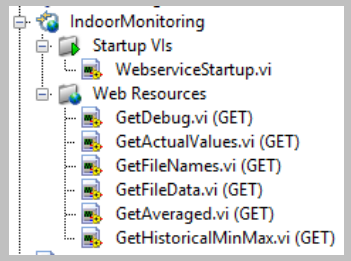
Agenda

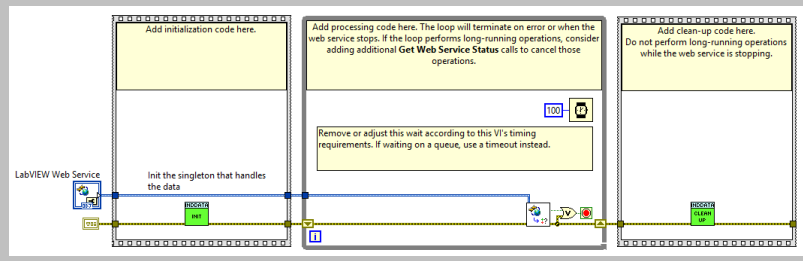
- Why did I start this project
- What is the setup
- How do we program LabVIEW on a Raspberry Pi
- How to interact with your application on the Raspberry Pi
- Build a LabVIEW NXG Webvi
- Host the Webvi on a SystemLink server

VI Technologies

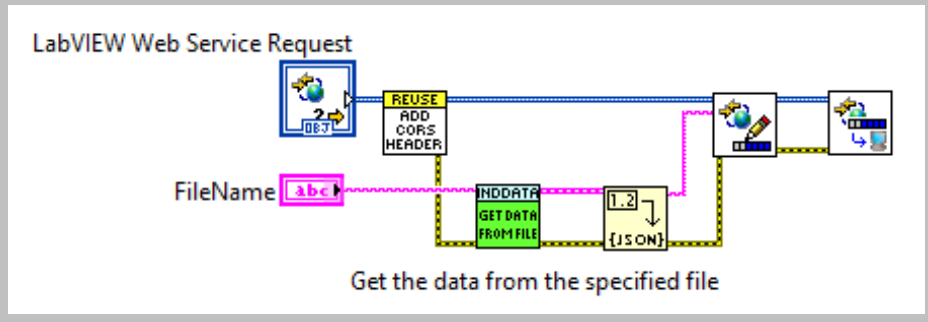
IndoorMonitoring WebService

- Right Click on target > New > WebService
  - Name space and project folders are created
- Startup VI's:
  - Execute when WebService starts
  - Init of the IndoorData Singleton (no refs)
    - Service starts before Application
- WebResources:
  - Your API
  - VI executes when URL is called
  - VI can have parameters (Ex. Filename)
  - Parameters are in the URL





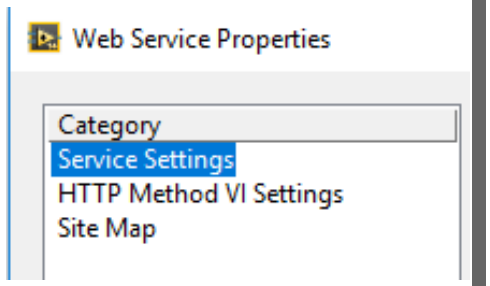
http://"IPAddress":8002/WebServiceName/VIName

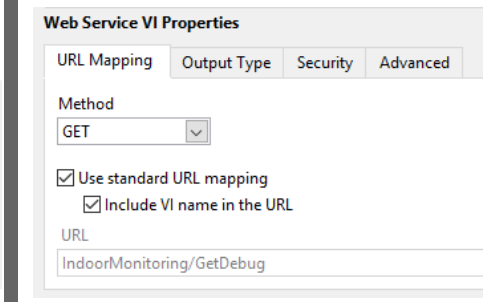


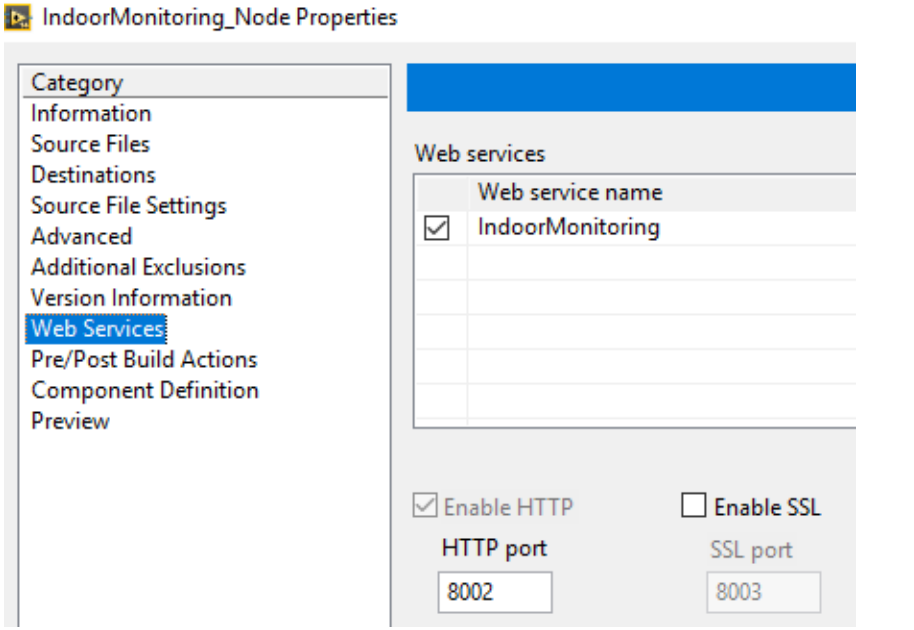
…IndoorMonitoring/GetFileData?FileName=20190519.txt

IndoorMonitoring WebService

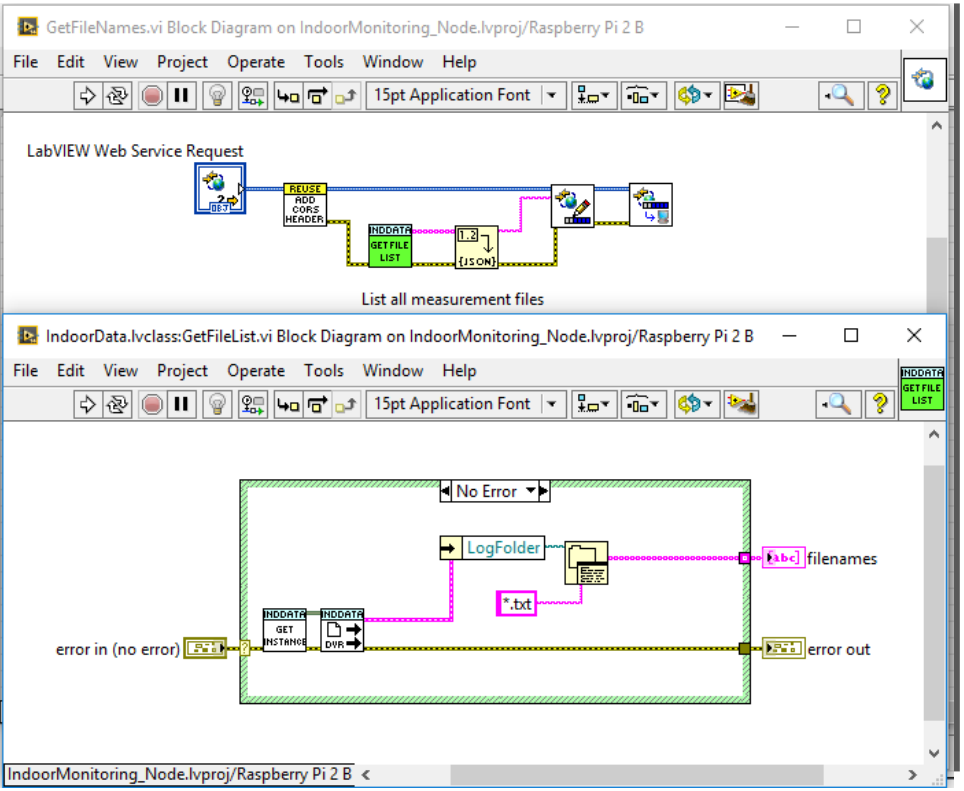
- Configuration:
  - Right click Web Service > Properties
  - Service Settings (name, version, …)
  - HTTP Method VI Settings
    - GET/PUT/POST/DELETE
    - Output Type: Terminal / Stream
    - Security
    - Advanced: Load VI @ startup/keep in Memory
  - Site Map
    - List of all Methods + URL (without parameters)
- Deployment:
  - In build spec > Web Services
  - Enable your webservice
  - Configure ports

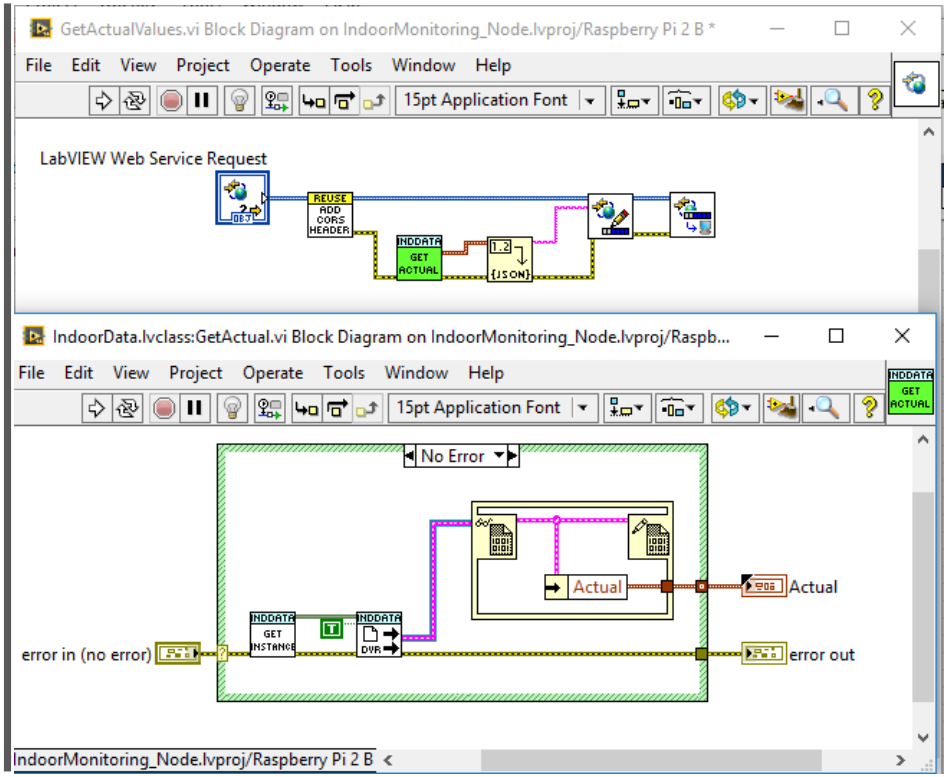






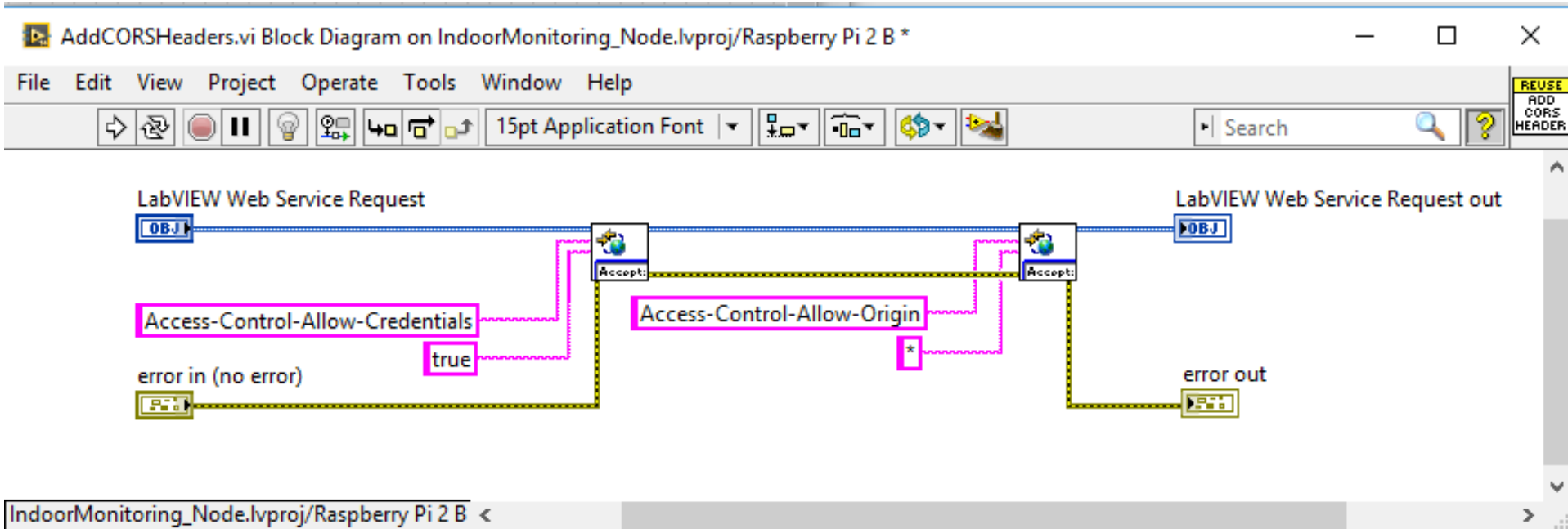
Examples of WebResources

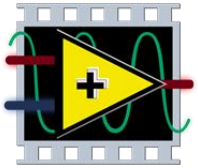




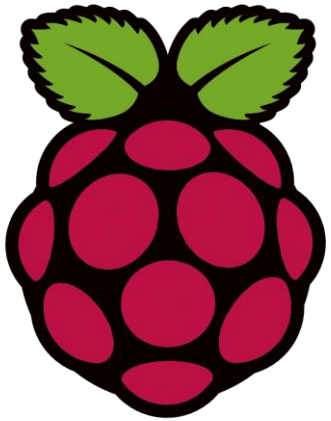
Add CORS Header

- CORS = Cross-Origin Resource Sharing
- allows a Web Service VI to respond to HTTP requests from a different server than where it is hosted.
- WebService = RPI >< Dashboard = SystemLink Server





LabVIEW



LabVIEW NXG

Agenda

- Why did I start this project
- What is the setup
- How do we program LabVIEW on a Raspberry Pi
- How to interact with your application on the Raspberry Pi
- Build a LabVIEW NXG Webvi
- Host the Webvi on a SystemLink server

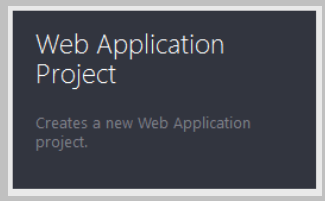
VIT echnologies

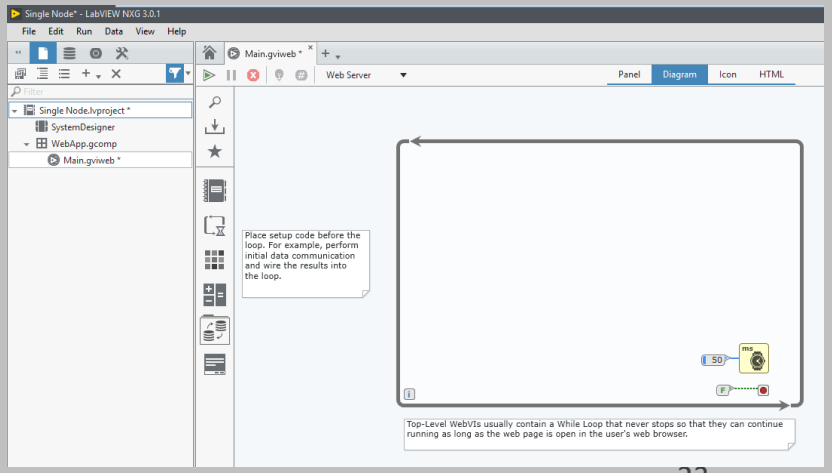
LabVIEW NXG WebVI

- LabVIEW NXG (3.0.1)
- NXG WebModule (NI Package manager)
- Projects > Web Application Project
  - Creates a LabVIEW NXG Project
  - Adds a WebApp.gcomp
  - Adds a main.gviweb
  - Not all functionality is supported in the webmodule…(yet?)

LabVIEW NXG

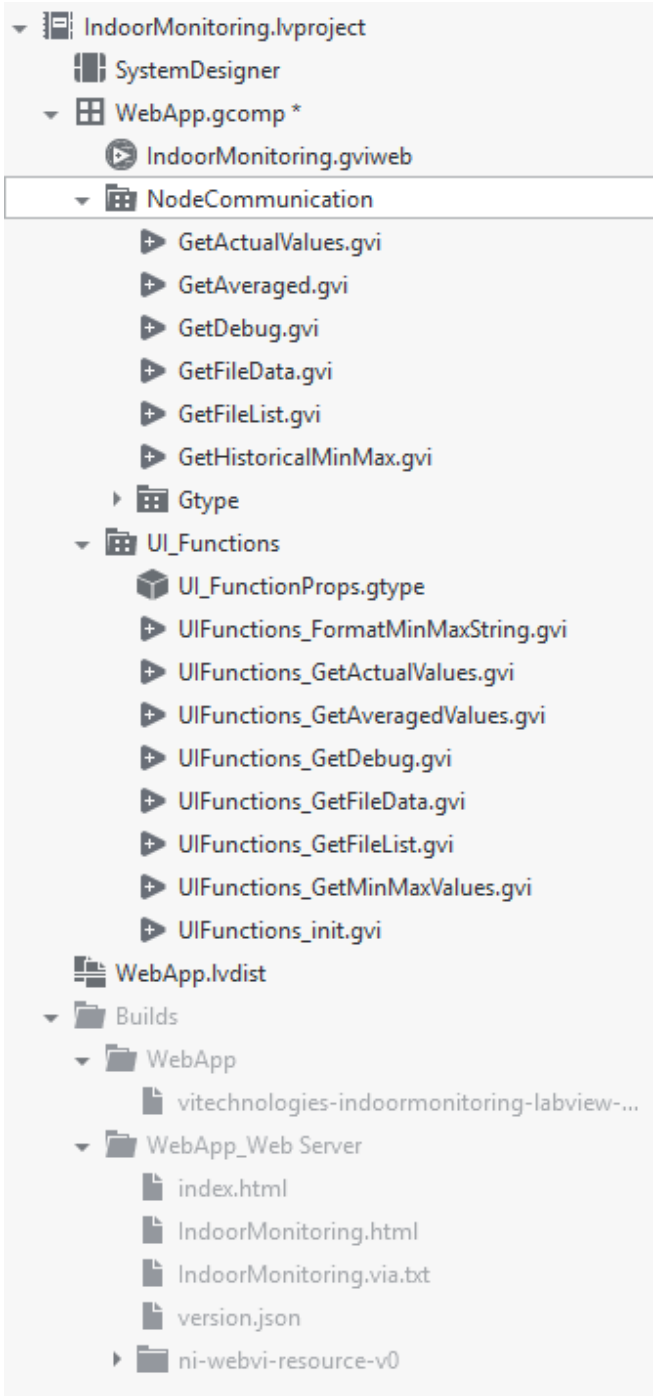
NXG Web Module



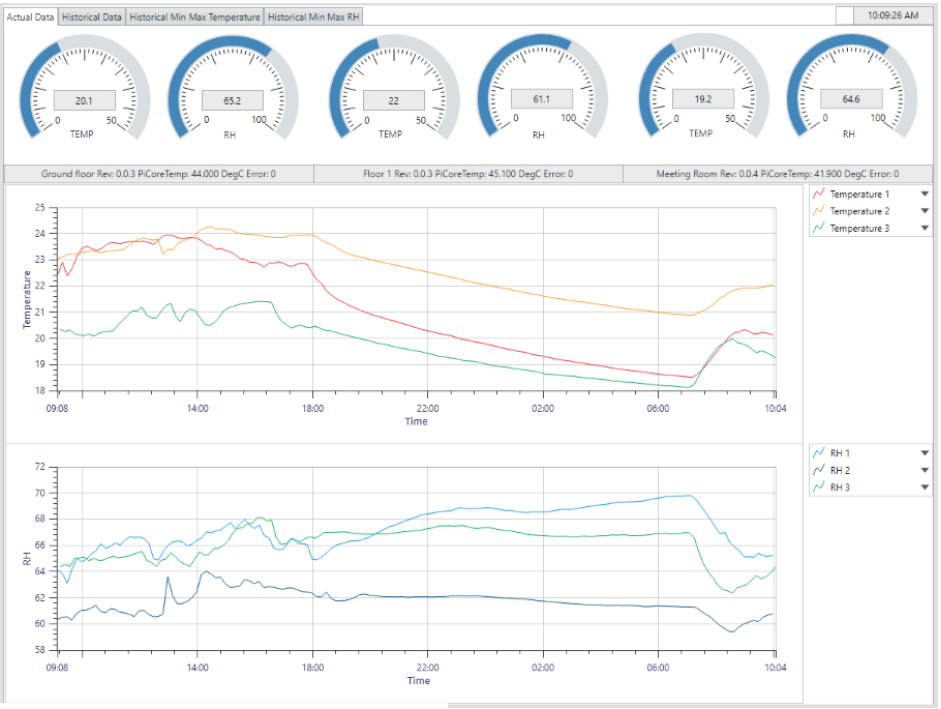


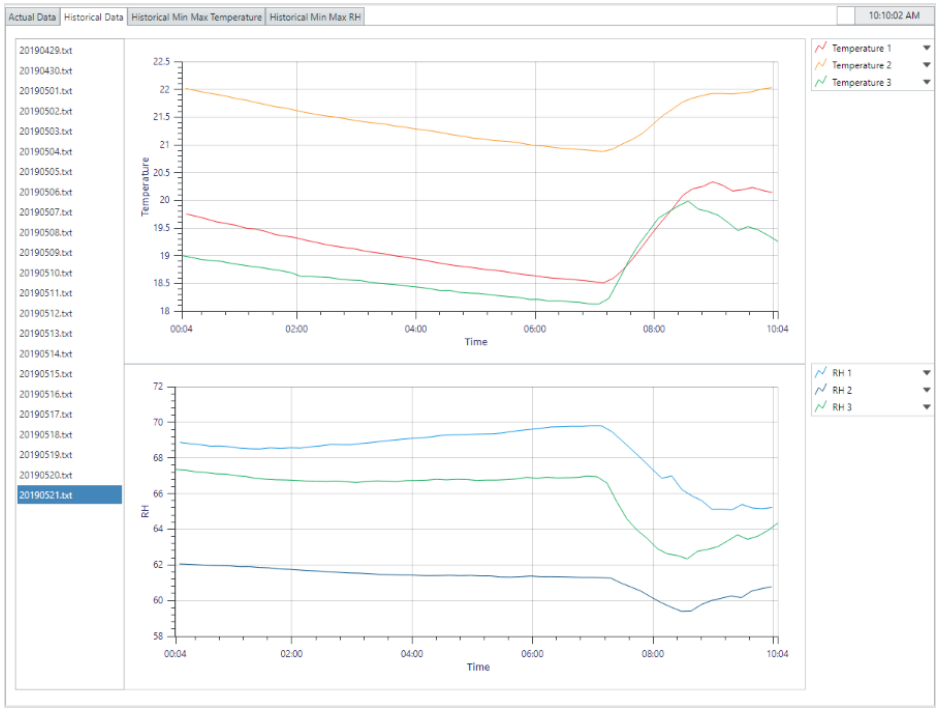
IndoorMonitoring Web App Project

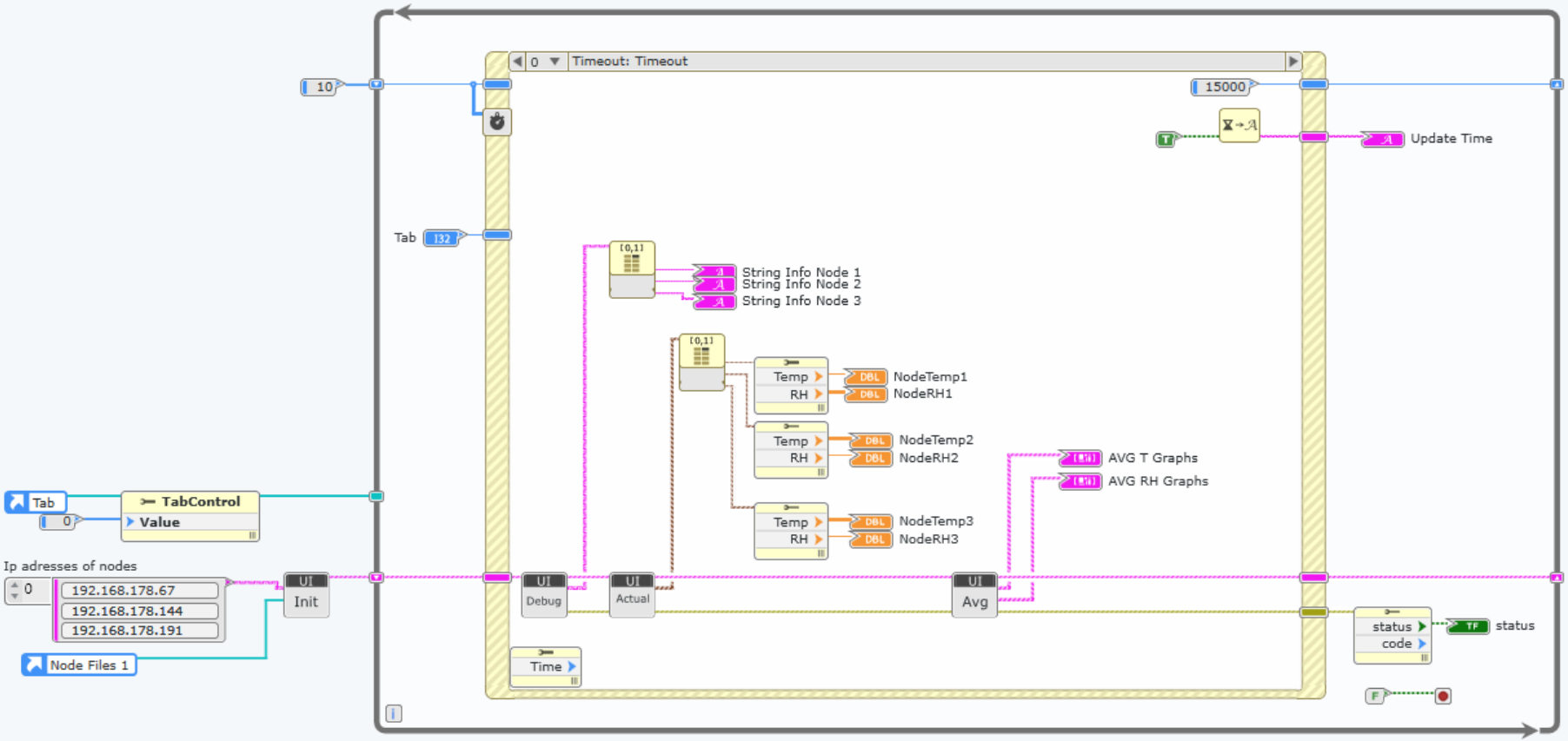
- WebApp.gcomp:
  - All items related to the Webvi
  - IndoorMonitoring.gviweb = main.vi
  - Namespace for NodeCommunication
    - Classes are not supported
    - Work around with namespace
  - NameSpace for UI_Functions
    - Classes are not supported
    - Work around with namespace
- WebApp.lvdist
  - The buildspec for my Webvi
- Builds



IndoorMonitoring WebVI UI

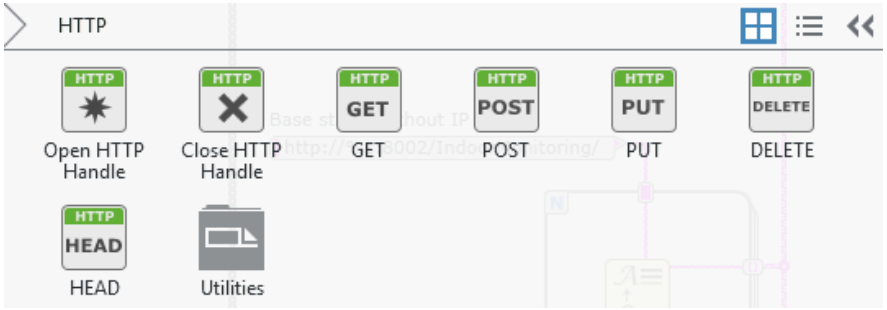


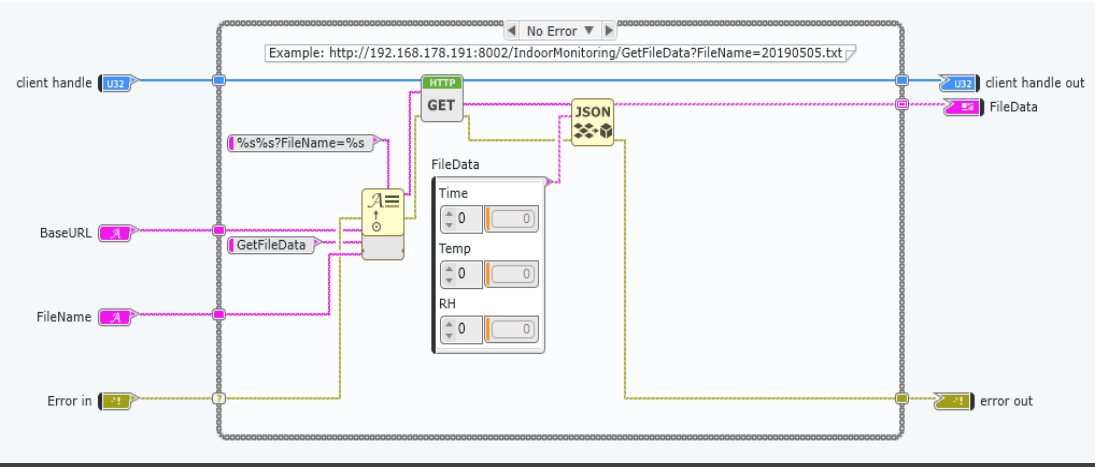


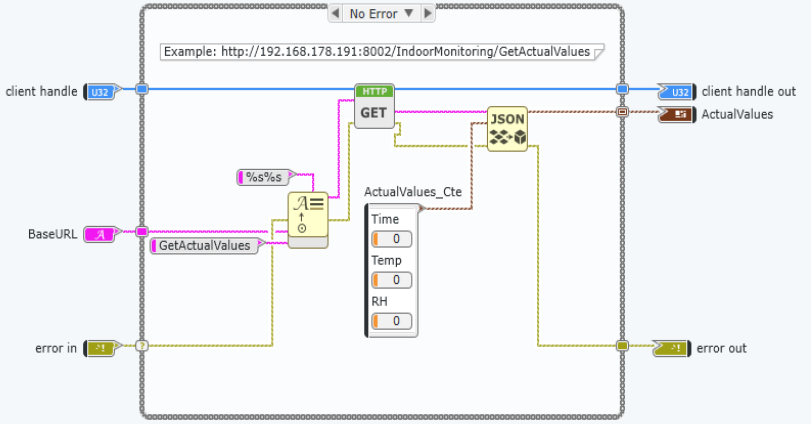


IndoorMonitoring WebVI

- Init prior to while loop
- While loop with event structure
- In Timeout update actuals
- Act upon events (Tab ctl, File List ctl)
- UI Functions: use NodeCommunication vi's to get the data

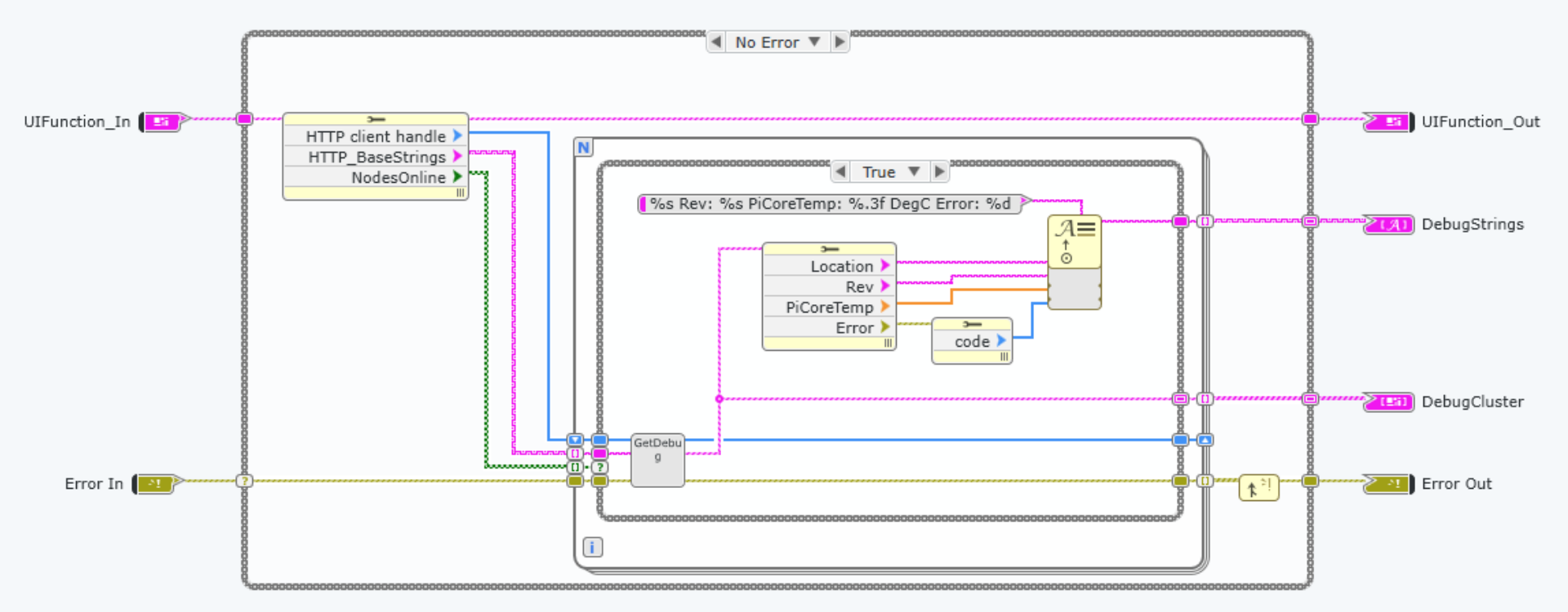






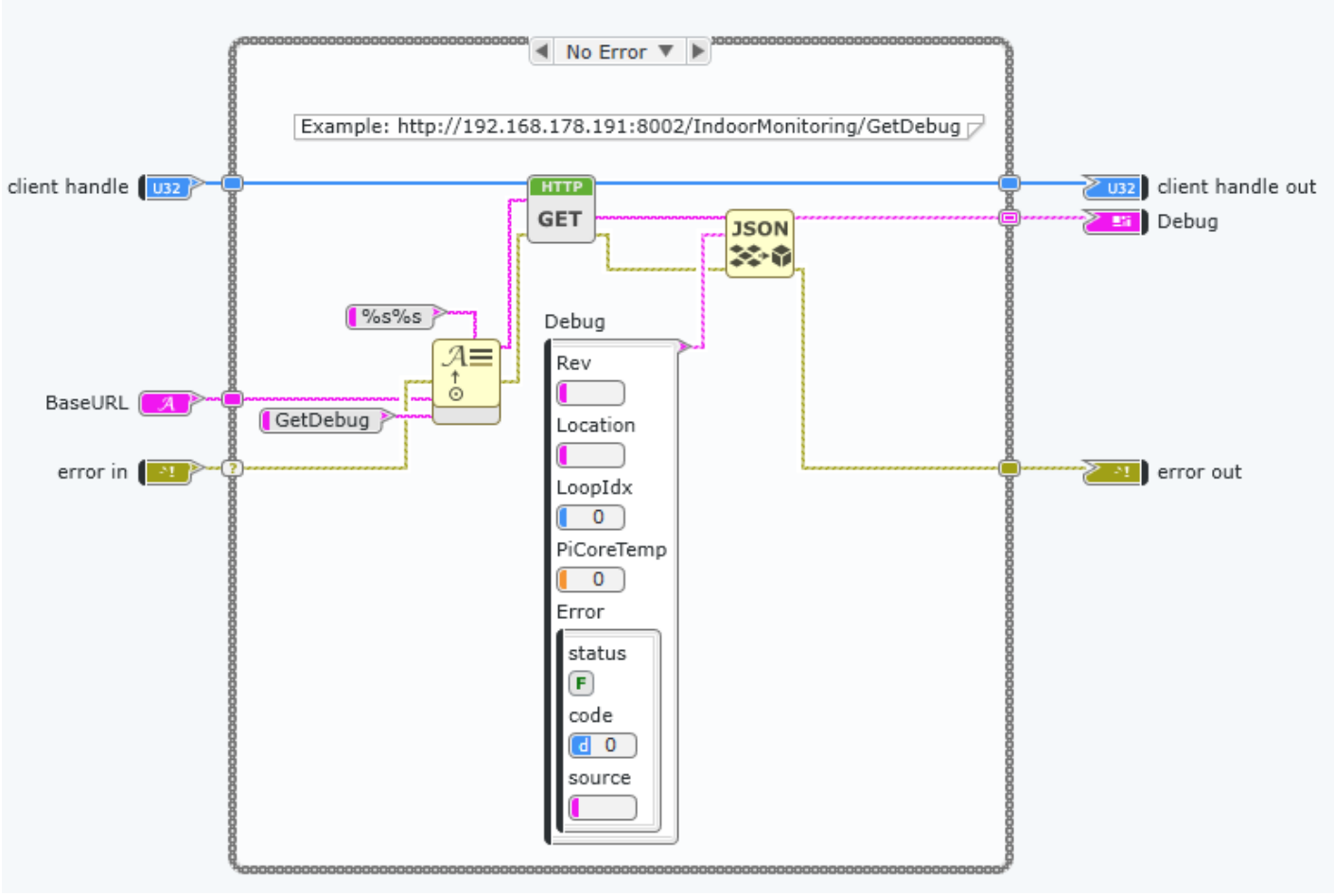
Data from nodes into Webvi

- HTTP Palette (Open, Close, Get, Post, …)
- URL Formatting (Webservice of RPI Node)
- Format reply into data structure



IndoorMonitoring UI GetDebug

- Unbundle cluster (Cluster properties)
- For every online node …
- Get the debug data
- Format debug data into string and ouput cluster

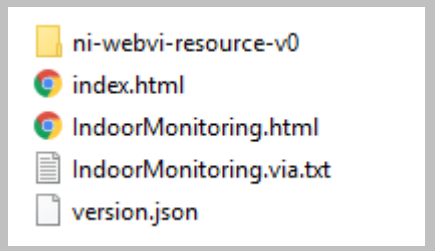


IndoorMonitoring GetDebug

- Format BaseURL to webservice API GetDebug
- Use the client handle for HTTP GET
- Unflatten from JSON
- Debug Data on terminal

Hosting an NXG WebVI

- Build a Webvi:
  - Output will give you an HTML page
  - And some resource files
- Apache Webserver on Raspberry Pi:
  - 2 commands to install
  - Copy output from LabVIEW NXG to your Raspberry Pi /var/www/html/ folder
- SystemLink Server:
  - Can host a Webvi
  - Documentation needs improvement

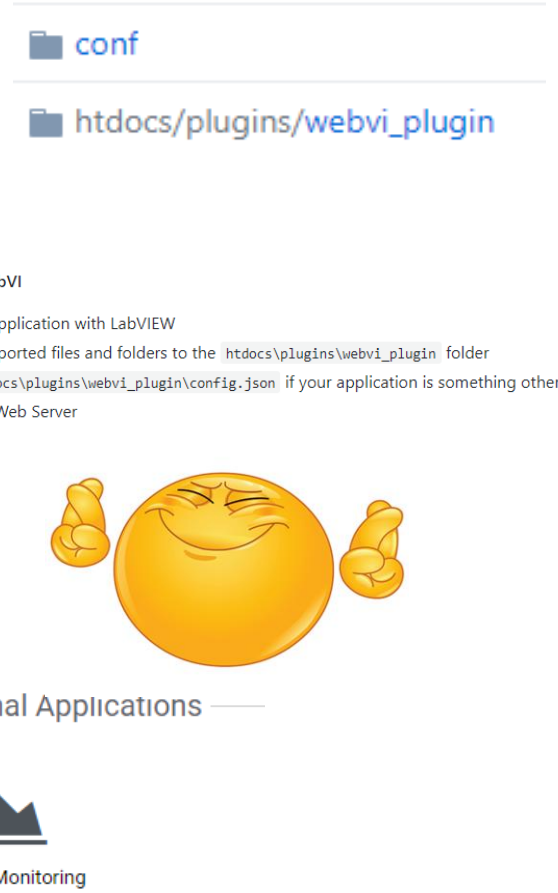



```
sudo apt-get update
```

```
sudo apt-get install apache2 -y
```

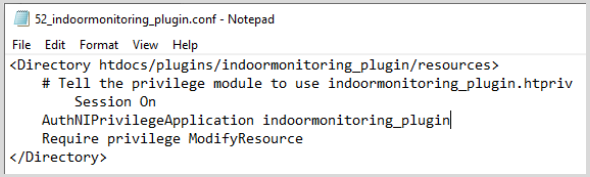
NXG WebVI on SystemLink Server

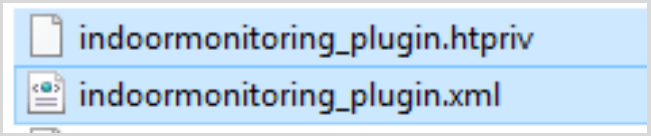
- A little documentation on [github]:
  - Install SystemLink Server (NI Package Manager)
  - Download the files from Github
  - Copy your Webvi to the server
  - Modify the files
  - Fingers crossed

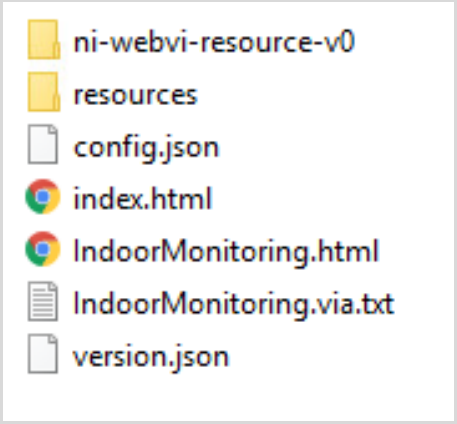


NXG WebVI on SystemLink Server

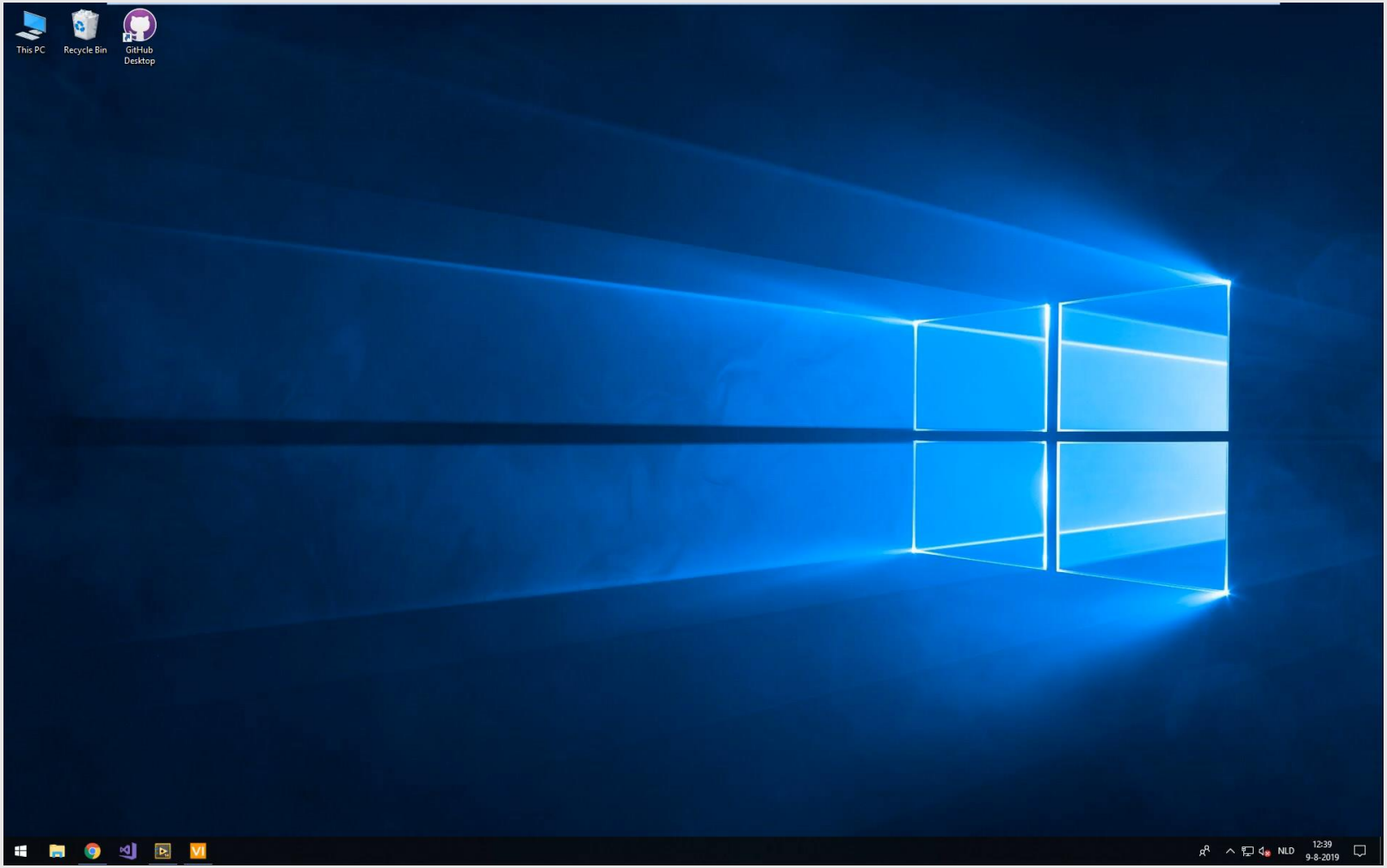
- C:\Program Files\National Instruments\Shared\Web Server\conf\conf.d
  - Create a config file Example: 52_indoormonitoring_plugin.conf
  - Update the file, should point to correct htpriv file
- …\Shared\Web Server\conf\htpriv.d
  - Xml file: adjust name
  - htpriv file: change filename
- …\Shared\Web Server\htdocs\plugins
  - Folder for your plugin/Webvi (indoormonitoring_plugin)
  - Copy the files from github here
  - Copy your Webvi here
  - Adjust
    - resources/json/locales/en.json
    - Resources/css/indoormonitoring_plugin.css

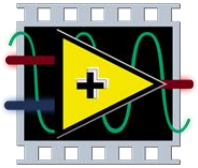




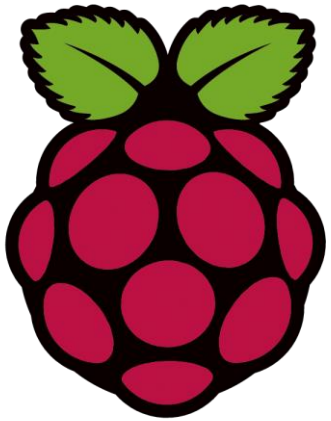


NXG WebVI on SystemLink Server





LabVIEW



LabVIEW™NXG

Agenda

- Why did I start this project
- What is the setup
- How do we program LabVIEW on a Raspberry Pi
- How to interact with your application on the Raspberry Pi
- Build a LabVIEW NXG Webvi
- Host the Webvi on a SystemLink server

VI Technologies

Indoormonitoring:
LabVIEW on a RPI & NXG WebVI Dashboard

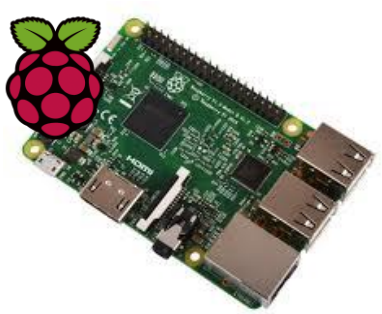
- LabVIEW on a Raspberry Pi: Makerhub
- Indoormonitoring: I2C sensor, singleton class for data, Webservice API
- LabVIEW NXG WebVI: HTTP API to access the webservice
- WebVI can be hosted on a SystemLink server

LINX

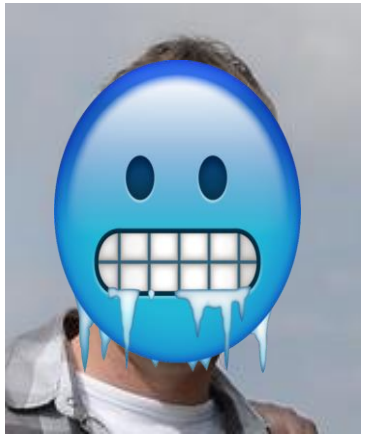
LabVIEW MakerHub

LabVIEW NXG

NXG Web Module



Indoormonitoring: Conclusions



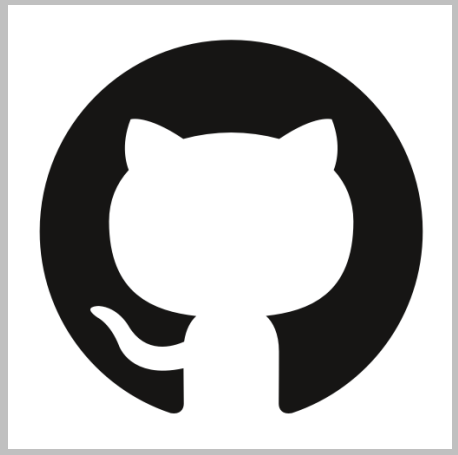
Links

Tools:

- Win32DiskImager (To write an image to uSD)
- WinSCP (Copy Files to RPI)
- VNC Viewer (Open a UI to RPI)
- Putty (SSH to RPI)

Github repo's:

- NI WebVI's
- NI SystemLink
- Wim Tormans



Questions