WildlifeSystems - biodiversity technologies

Ed Baker

2022-08-17

Contents

About

This book explains the technologies developed as part of WildlifeSystems and how they can be implemented in real-world scenarios.

Chapter 1

Biodiversity Technologies

What are *Biodiversity Technologies*?

1.1 Structure of Wildlife Systems

1.1.1 Overall Philosophy

1.1.2 Packages

Chapter 2

Sensor Networks

Chapter 3

Environmental Sensors

3.1 How sensors work

3.1.1 Temperature

3.1.2 Humidity

3.1.3 Air Pressure

3.1.4 Gases

3.1.4.1 Heated Gas Resistance

Chapter 4

Sensors in WildlifeSystems

The WildlifeSystems platform comes with support for some popular existing environmental sensors, although there are many on the market and the range available is subject to constant change. The modular nature of WildlifeSystems allows for new sensors to be easily integrated if the need arises.

4.1 Sensors included in the base system

The Raspberry Pi does not come with environmental sensors, however there are several onboard sensors that are used to monitor the operation of the hardware, to prevent crucial components from overheating, including the temperature of the CPU and GPU chips. WildlifeSystems provides access to these sensors through the `sensor-onboard` package, as well as providing some *software sensors* that report the free memory and free SD storage available. These can be useful for detecting and resolving possible issues on a sensor node before a serious problem arises.

4.2 Installing sensor support

Support for sensors is installed as part of the node installation process, however it is possible to install the `sensor-control` abstraction layer onto any Raspberry Pi system using the command below.

```
wget -O - https://raw.githubusercontent.com/Wildlife-Systems/sensor-control/main/install | sudo b
```

This will install the `sensor-control` and `sensor-onboard` scripts into the system, as well as installing a small of number of supporting packages if they are not already installed.

4.3 Reading data from a sensor

The sensor read command, `sr`, can be used to read sensor data.

```
sr onboard
```

This will give a JSON string listing information about each sensor, and the current reading. This information can be presented in a more human readable form by piping the output to the program `jq`, a command line JSON processor.

```
sr onboard | jq
```

4.4 The sensor reading process

The sensor reading process in WildlifeSystems has five steps.

1. The `sr` command identifies which sensor script to route the request to.
2. The sensor script calls the `sc-prototype` script to obtain a template ("prototype") of the JSON request.
3. The sensor script access the relevant sensor(s) and populates the values in a template for each sensor reading, before returning a JSON array of populated readings to `sr`.
4. The `sr` script populates additional information for each reading, providing the `node_id` and a `timestamp` for each.
5. `sr` returns the finalised JSON array to the user.

4.5 Installing new sensors

Chapter 5

Power Management

Power management on the Raspberry Pi is useful when deployments are made that are powered by batteries and/or renewable sources such as solar power that are intermittent.

In addition, there are small environmental benefits on consuming less power on systems which have continual grid power.

The `pi-pwr` script can be used to turn off unused functionality, either always or just when it is not required.

5.1 Installation of power management tools

The power management tools are installed as part of the node installation process, however they can be easily installed independently on any Raspberry Pi system.

```
wget -O - https://github.com/wildlife-systems/pi-pwr/raw/master/install | sudo bash
```

5.2 Turning funtionality on and off

5.3 Considerations

Disabling network functionality (WiFI / Ethernet) will prevent the node from communicating until either the functionality is turned back on or the Raspberry Pi is restarted. If disabling all connectivity is desired periodically then the functionality to turn these systems back on must be scripted.

Chapter 6

Developer Guidelines

6.1 Documentation

There are three main kinds of documentation in the WildlifeSystems project.

1. **Code comments**: Primarily for future you (or someone similar) to get to grips with your code. *Why does this code do this?*.
2. **Package documentation**: Describes what your package does, how it interacts with the larger system. *What will this package do for me?*
3. **Overall project documentation**: This manual. *How do I use the entire system for my research?*