**Setting up the Raspberry Pi based field sensor monitor**

August 2016

# Contents

# 1 INTRODUCTION

This document describes how to set up and use the field monitoring system based on a Raspberry Pi base station and Moteino/Arduino based loggers. It provides a description of the hardware used and how to set up specific parameters in software, such as logging frequency.

In this document, the term "controller" refers to the Moteino which is attached to the Raspberry Pi. It receives data from "loggers", which are the independent Moteinos with photodiodes attached. The Raspberry Pi takes pictures and is turned off and off by the controller. The term "base station" refers to the combination of Raspberry Pi and Controller, because they work together.

The network operates on a star topology, with individual battery powered loggers reporting back to a base station. Loggers read from their photodiode arrays only when they are told to by the controller. The controller tells the loggers when to read from their sensors, and also turns the Raspberry Pi on and off. Turning the Pi off when it's not needed results in a dramatic power saving. Like the loggers, the controller is a Moteino microcontroller with a radio attached. It's necessary to set up radio parameters so that the right loggers are talking to the right controller. These parameters are set out and explained in this document.

The Raspberry Pi acts as a data logger and camera operator. At each logging interval (which must be a multiple of 5 minutes) it is woken by the controller. The controller sends any data received from the loggers to the Raspberry Pi. Once this data has been received, the Pi captures and saves an image before automatically shutting down.

## 1.1 What you need

Arduino IDE

# 2 HARDWARE

## 2.1 Base station

## 2.2 Logger

# 3 SOFTWARE

## 3.1 Raspberry Pi

The SD card image provided has everything already set up, but there are some more details here if you need to change anything. If you just want to set up

a new Pi with the provided image, go to **Setting up the Pi: Simple**.

The Raspberry Pi is set up so that it automatically runs a script when it starts. This script listens for any input from the controller and then captures and saves an image using the camera. Then it shuts down the pi. This script is located in */home/pi/scripts* and it runs from */etc/rc.local*. Scripts run from *rc.local* run as the "root" user (similar to Administrators in Windows). You can look at this file by typing *cat /etc/rc.local*. The Raspberry Pi is turned off and off by the controller; it turns the Pi on, waits for it to complete its tasks, then turns it off.

#### 3.1.1 Setting up the Pi: Simple

#### 3.1.2 Setting up the Pi: Complicated

## 3.2 Controller

## 3.3 Logger

The logger is the Moteino with the light sensor array attached. It runs from 2 x AA batteries and has a low power radio to send data back to the controller. The controller that the logger is sending data back to must have the same *NETWORK_ID* as the logger, or they can't communicate.

When setting up a new logger, there are a few things that you need to change, and a few things that you can change if you want to. As a minimum, you need to:

- Check that the logger is on the right network, so that it can communicate with the right controller
- Check that the logger's *LOGGER_ID* is unique on it's network
- Check that the logging frequency is correct

#### 3.3.1 Radio parameters

**LOGGER_ID** The *LOGGER_ID* is the unique number of the logger. All loggers with the same *NETWORK_ID* need to have different *LOGGER_ID*'s. This is partly so that when you look at the resulting data, you will know what logger it came from. It also allows the radios to identify each other, and to prevent clashes between loggers.

The logger ID is transmitted to the controller along with the data from the logger. Because there is a limit to how much data can be transmitted at once, there is a limit to how big the *LOGGER_ID* can be, because it is only allowed to take up one byte, or eight bits. The highest number you can represent with eight bits is 255. So, including zero, we can have up to 256 unique *LOGGER_ID*'s on

one network (networks work the same way). One of these numbers (the number 1) is actually taken by the controller, and the number 255 is used as a "broadcast", meaning that all loggers on the network can see data transmitted to 255.

So, when you are setting up a new logger, set the logger ID to soemthing greater than 1 and less than 255, and make sure that no other loggers with the same *NETWORK_ID* have the same *LOGGER_ID*.

**NETWORK_ID** The *NETWORK_ID* is used to identify which loggers are on the same network, i.e. which loggers can talk to each other. If you have a Raspberry Pi set up in one field, with a controller with a *NETWORK_ID* of 100, all of the loggers that need to talk to that controller must also have a *NETWORK_ID* of 100. If you set up another base station in a different field nearby, you probably don't want the loggers in the first field to also be transmitting to that controller.

So, each controller has a unique *NETWORK_ID* (like *LOGGER_ID*'s, these must be greater than 1 and less than 255), and all loggers that send data to that controller have the same *NETWORK_ID*.

**GATEWAY_ID**

**FREQUENCY**

**ENCRYPTKEY**

**ENABLE_ATC**

### 3.3.2 Logging parameters

**NUM_PHOTODIODES**