Plant Phenotyping Camera System Implementation Notes

(NCHAIN Project 2018)

Table of Contents

1 Introduction

- **Overview:** These implementation notes aim to guide others towards implementing a high throughput plant phenotyping system from start to finish; involving the components, processes and design for a using embedded computers with cameras in a system for capturing images of plants for analyzing their growth over time.

Comments are in grey, slanted text.

- **Background Story:** The system was used in 2018 by the Institute of Molecular Biology & Genetics, Aarhus University, during project NCHAIN, in search for a better understanding of the organic nitrogen chain.

- **Design Goals:** This is a durable and reliable system for producing several images of 1800 plant/pots every day over a period of 5-6 months, with the possibility to monitor the ongoing process. The method aims to be as reproducible as possible, sticking to widely available software and hardware components. A plan for ensuring the integrity and safety of the image files is included; all while trying to keep costs at a minimum. The final is to be able to analyze images and be able to produce consistent data about the plants.

- **Data Transfer Timing:** The 180 Raspberry Pi 3 Model B computers monitoring 1800 plants, proved to be more than capable of transferring images over the 100 Mbit network, as long as they did not transfer all at the same time. Therefore we implemented an asynchronous scheduling method for taking pictures and transferring them to our control server, running transfers from few cameras simultaneously. Since the broadband connection would only transfer at around 300-500 kB/s, and for easier file management, we decided to compress every timeslot of images into a timestamped archive.

- **Automated Backups:** Additionally to these picture taking events, while idling; the system will perform separate backup routines at hourly intervals. This includes storing image archives as an extra backup on the server running at the greenhouse, in case of a connection loss to our remote scheduling and storage. This way, we were better protected against loss of data points.

- **Software:** The operating systems of the rpi and the greenhouse server were both configured to boot and initiate production without human intervention. This provided for resiliency to downtime during power outages (several power outages occurred throughout the 2018 experiment).

2 System Details

- **General Description:** The computer components include 180 raspberry pi with cameras + 1 server which is acting as a central control unit. These 181 computers are connected to each other via LAN cables and network switches. Everything is mounted in the ceiling 2 meters above the tables where the plants are.

2.1 Physical System Architecture

- **Server:** The server used for controlling all the cameras in the greenhouse was an 'old' Sun x2200 with 2 GB of ram, a 500 GB consumer grade hard disk for the operating system and a 1TB consumer grade hard disk for backup images. The server sits inside of a slightly ventilated tent made of reflective mylar blanket, to keep the temperature constant and protect against the environment.

- **Cameras:** The Pi Camera V2 is featuring the Sony 8MP IMX219 image sensor, which we found to have both an acceptable resolution and decent low light performance.

During the course of the experiment we had 10% of the camera modules fail; implying a worsening of image quality, caused by moisture, lens being out of focus, debris on the lens, and more. 25 out of 180 camera modules were to be replaced.

- **Router:** For internet connectivity we used a mobile broadband router

We found the connectivity in the greenhouse to be subpar(slow transfer rates); which was likely due to the location on the country, and the greenhouse inhibiting signals.

- **Network:** The network is comprised of 46 100Mbit switches connected in series, with LAN cables extending out to each camera. The control server is connected to this network, while also being connected to the internet on ‘a seperate interface. This allows for retrieval of images, isolation of the cameras to the internet, and transfer away from the greenhouse.

We used 100Mbit switches, which were adequate for our purposes. If the number of cameras or image sizes were to increase, more bandwidth might be required.

2.2 Component Prices

The camera system was constructed from inexpensive Raspberry Pi embedded computers, consumer-grade network equipment, and generally available cheap construction parts.

- **Parts and prices**: A general list of materials used for the construction of the Raspberry Pi camera setup and corresponding prices*.

Component	Count	DKK Prices*
Electrical Components: - Electrical Mains Wiring and Sockets - Power Strips	- 220v plugs, power strips, installation cable - mains wiring 300m - USB cables - USB power supplies	6500 800 500 3400
Network Components - LAN cables, Switches	- 46 8-port switches - network cables (240 pcsvarious length) -	4500 6000
Miscellaneous - Construction parts - Spare parts - Tape, cable ties	Tape Internet Miscellaneous Utility Metal U-profiles	150 2500 500 8100
Raspberry Pi Based Camera Components: - Clear Plastic Cases - Mainboard (Raspberry Pi) - SDcards - Camera Module	Raspberry Pi V3 B 1GB ram 1.2 GHz (181) Pi Camera module V2 (200) cases (75+86 185 Micro SD cards (bad quality) 145 Micro SD cards (extra)	50000 36500 4700 4800 9500
TOTAL		138450

*Prices are indicative

3 How-To : Camera Software

This information should get you from empty microSD card to functioning network connected camera step by step.

3.1 Camera Installation Requirements

- microSD card large enough to fit the raspbian operating system (~2GB and up)
  (any data contents will be obliterated during the installation process)
- A computer with a linux system (if you want to use a specific way of cloning the camera systems).

3.2 Camera Installation

- Download the raspbian image from : https://www.raspberrypi.org/downloads/raspbian/
- Follow the official guide on how to install the system onto a microSD card: https://www.raspberrypi.org/documentation/installation/installing-images/README.md

You don’t have to truncate/size the image to the microSD in case the card is larger , so skip this step if you prefer as it will save you time copying the image later on! - Less data is faster to copy,

3.3 Camera Configuration

Once you have a working installation, you need to set unique IP addresses for your cameras.

1. **Setting Unique IP addresses for each camera:** Open the first partition on the microSD, and edit the file cmdline.txt, setting the IP address (look this up if in doubt)

2. **Remote Control:** Set up SSH login without password to the raspberry pi system.

There are many guides available online for doing this on linux-based systems. (you want to be able to open an SSH connection to the raspberry pi without having to provide a password)

3. **Fault tolerance:** Make the system read-only to prevent wearing out the microSD cards or risking data loss on power failure.

4. **Cloning:** Now we have installed and configured one camera module and we are ready to clone it to produce another functional camera module. I give you two options for cloning the card:

    a. You can use this script to achieve cloning on a linux-based computer (with paranoid extra features for correct writing):

https://github.com/MarcClausen/NChain_project/blob/master/SDwriter.shThe script is checking the filesystems and makes sure everything gets written out correctly to the microSD card.

b. You can also make an image of the microSD yourself, and write that onto another card.

4 How-To : Server Software

4.1 Server Installation

Hardware-wise we recommend a platform that will be able to perform consistently e.g. enterprise hardware.

- **Operating system:** For the computer/server controlling the cameras, we recommend a robust operating system, which is able to run for a long time without breaking itself (Slackware, CentOS etc.), and preferable one you can operate. - We used Slackware

4.2 Server Configuration

- **Network:** Our server was network-connected via 2 LAN interfaces. Interface 1 to a mobile broadband router. Interface 2 to the switches connected to the cameras.
- **Camera script:** For automating the picture-taking process. A single bash-script was used. The script can be viewed here (including further documentation) : https://github.com/MarcClausen/2017_2018_NCHAIN
- **Automatic operation:** When you boot the server, network should be working, and the script should be set to run as a user other than root.

5 Economic Considerations

5.1 Flash Storage MicroSD Cards

We recommend that you buy the microSDs from a reputable dealer/source and get a warranty, as faulty microSDs will be more costly in man hours and increase the risk of losing mission critical data opposed to getting microSDs of sufficient quality to begin with. We started off using 180 SDcards ordered directly from a generic manufacturer to keep costs low, but many of these cards were to corrupt over time and would need replacements. We then purchased microSDs from a more reputable source (Toshiba Exceria 16GB), which largely reduced the number of corruption incidents.