Project Progress

Aleksandar Krastev (s0833784)

June 18, 2013

1 June 17, Sunday

- Downloaded Raspbian Linux. It is a flavour of Debian Linux that is optimised for the Raspberry Pi. Its packets are compiled to use hard floating point arithmetic because the Raspberry Pi does not have floating point hardware to bring costs down. The operating system's image was downloaded from `http://www.raspberrypi.org/downloads`.
- Followed instruction at `http://elinux.org/RPi_Easy_SD_Card_Setup#Using_the_Linux_command_line` to install an operating system on the 3 SD cards.

2 June 18, Monday

- Tested the 3 Raspberry Pi that their operating systems booted and ran as expected. Updated the systems and expanded the SD card partitions to fill the entire space on the SD cards (3.9Gb). Renamed the hostnames of the devices pi0, pi1, pi2. Marked the devices and SD cards with numbers 0, 1, and 2.
- Connected the 3 devices to a switch (given to me by my flatmate). Then connected the switch to the our flat's broadband ADSL router. I reserved the following IP addresses for the devices - 192.168.1.10, 192.168.1.11, 192.168.1.12 - using the RasPi's MAC addresses for identification. The devices successfully connected to the router and were assigned their reserved IP addresses. They could ping websites (internet connection) and could ping each other (local connectivity).
- Installed a web server (apache2), PHP (php5), and an SQL database (sqlite3) on pi2. The web server was running successfully and could be assessed through the LAN (Local Area Network).

- The idea is to code in Python to store data in a sqlite database that could be accessed by PHP to display data on the website that could be accessed remotely. Further plans include setting parameters through the website so that they could affect the remote system. Have to think about (explicit) transactions in the database so that both sides of the system can read/write without race conditions or data corruption, i.e. need to find means for synchronisation.

- Wrote a small PHP script that could be found at `192.168.1.12/status2.php` on the LAN and runs on pi2. The script does checks if the a SSH socket (number 22) can be opened on all three devices and displays if the devices are ON or OFF at the web page.

```
<?php
$pis = array(
        '192.168.1.10',
        '192.168.1.11',
        '192.168.1.12'
);
$data = '<html><body>';
for ($i = 0; $i < count($pis); $i++) {
        $sock = @fsockopen($pis[$i], 22, $errno, $errstr, 1);
        if ($sock) {
                $data .= '<p>pi'.$i.':␣ON</p>';
                fclose($sock);
        } else {
                $data .= '<p>pi'.$i.':␣OFF</p>';
        }
}
$data .= '</body></html>';
echo $data;
?>
```

- Created an image of the current configuration of the pi2 (working OS with web server) so that if something goes wrong the system can be restored fast. Compressed the image (using xz) so that it takes less space.

- Applied for a free Student Micro Account (5 private repositories) on GitHub. My account was upgraded and I created a private repository for my project at `https://github.com/sandio/raspi-rfid-tracking` (not accessible because it is a private repository). If Michael Rovatsos or Michael Anslow have profiles at GitHub can add them as collaborators. This repository will be used by me to record any progress made as well as to write my thesis. I think it is a good place because I can record different versions of my work. The repository will be used by the RasPis also. It will contain the source code of the project. Any changes will be documented and committed to the GitHub server so that all devices and collaborators have the most recent version of the project. I plan to tag a bundle of source code changes into versions (eg. v0.1) so that the project progresses through versions with added functionality or fixed bugs. I plan to use GitHub's issue tracker in order to record bugs and my feature requests.

3 June 19, Tuesday

- Created this project progress document.
- Came up with an idea that all devices can run the same setup. More specifically, each device will get data from its neighbours and compute a possible position. After 3 positions are computed independently one RasPi can act as an arbiter to decide which is the most probable position. This idea can be applied once one device can compute the tag's position based on the distance readings of the 3 RasPis.

Testing that references work [Stelzer et al., 2004].

References

[Stelzer et al., 2004] Stelzer, A., Pourvoyeur, K., and Fischer, A. (2004). Concept and application of LPM-a novel 3-D local position measurement system. *Microwave Theory and Techniques, IEEE Transactions on*, 52(12):2664–2669.