

# Design of Embedded and Intelligent Systems, Lab 2

Group 1, Harald Lilja, Anton Olsson, Simon Brunauer

September 2019

## 1 Introduction

This is a report for the second lab in the course Design of Embedded and Intelligent Systems, conducted in the autumn of 2019 at Halmstad university. The lab consists of 4 parts that are to be completed in order to pass the assignment. The following hardware is used in the lab:

- Raspberry pi 3
- SparkFun RedBot Mainboard - Arduino microcontroller
- Raspberry pi camera V2
- Host machine - Dell XPS15 running ubuntu 19.04

The lab was divided into the following parts:

**Part 1** Connect to the raspberry pi from the host machine using ssh

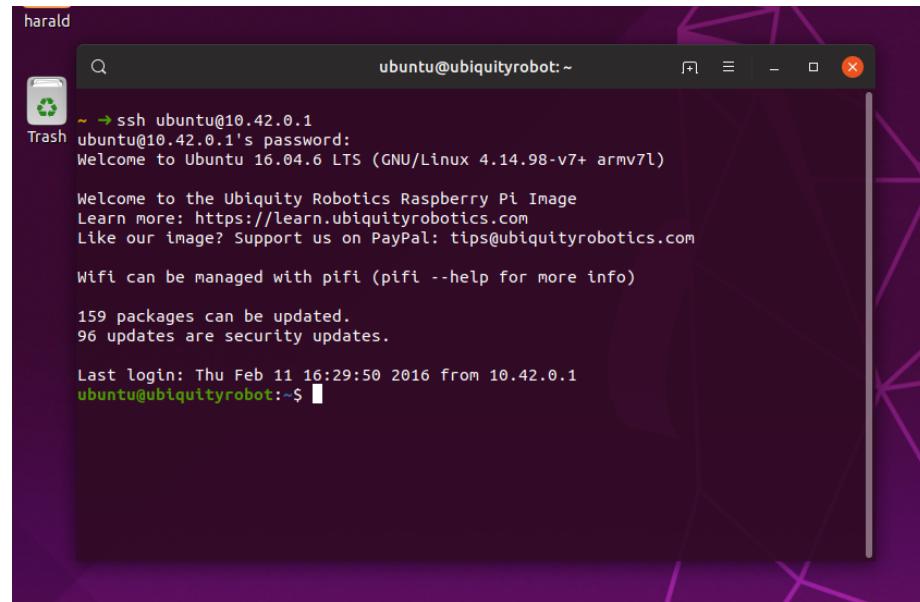
**Part 2** Write a program and run it on the arduino microcontroller

**Part 3** Send and receive messages using ROS on the raspberry pi

**Part 4** Take a picture using the camera on the raspberry pi

## 2 Connecting to the raspberry

In order to connect to the raspberry using ssh a mobile hotspot was setup using a mobile phone. The host device and the raspberry were then connected to the access point and a successful ssh connection could be established, shown in Figure 1.



The screenshot shows a terminal window titled "harald" with a dark purple background. The title bar includes a search icon, the terminal name "ubuntu@ubiquityrobot: ~", and standard window controls. The terminal itself displays the following text:

```
~ ➔ ssh ubuntu@10.42.0.1
Trash ubuntu@10.42.0.1's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.14.98-v7+ armv7l)

Welcome to the Ubiquity Robotics Raspberry Pi Image
Learn more: https://learn.ubiquityrobotics.com
Like our image? Support us on PayPal: tips@ubiquityrobotics.com

Wifi can be managed with pifi (pifi --help for more info)

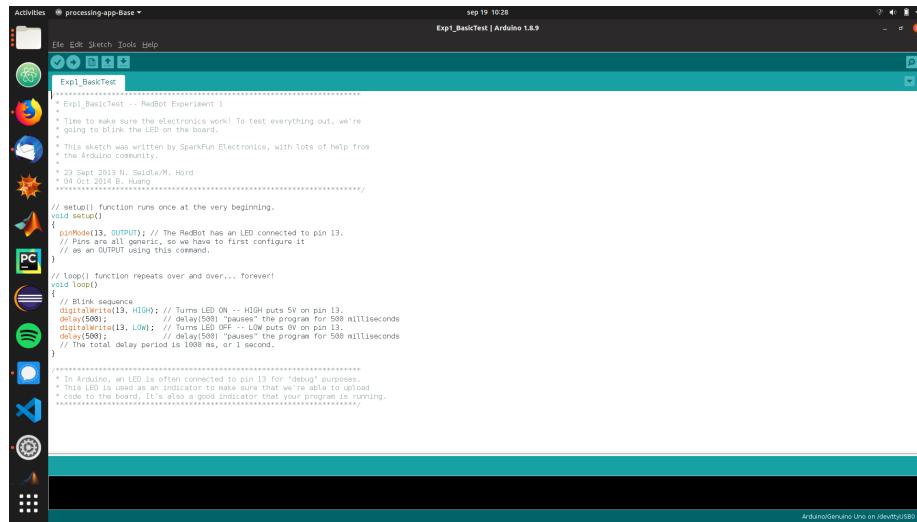
159 packages can be updated.
96 updates are security updates.

Last login: Thu Feb 11 16:29:50 2016 from 10.42.0.1
ubuntu@ubiquityrobot:~$ █
```

Figure 1: Established ssh connection

### 3 Arduino program

For the second task the arduino IDE was installed which is a development environment for arduino hardware. There was a library for the sparkFun redbot package that could be installed with prebuilt sketches for the microcontroller. One of these were a basic blink program which simply makes the LED on the controller to blink. This program was compiled and flashed to the controller.



The screenshot shows the Arduino IDE interface with the sketch 'Expt\_BasicTest' open. The code is as follows:

```
Activities ● processing-app-Base
File Edit Sketch Tools Help
Expt_BasicTest | Arduino 1.8.5
sep 19 10:28
Expt_BasicTest
Expt_BasicTest - RedBot Experiment 1
Time to make sure the electronics work! To test everything out, we're going to blink the LED on the board.
This sketch was written by SparkFun Electronics, with lots of help from the RedBot community.
- Last updated: Sep 19, 2013 by: Saldia/M. Hord
- Created: Sep 19, 2013 by: Saldia/M. Hord
- Version: 0.1
- Date: 04 Oct 2014 by: Hord
// setup() function runs once at the very beginning.
void setup()
{
  pinMode(13, OUTPUT); // The RedBot has an LED connected to pin 13.
  // Pins and all that jazz, so we have to first configure it
  // as an OUTPUT using this command.
}
// loop() function repeats over and over... forever!
void loop()
{
  // Blink sequence
  digitalWrite(13, HIGH); // Turns LED ON -- HIGH puts 5V on pin 13.
  delay(500);
  digitalWrite(13, LOW); // LOW puts 0V on pin 13.
  delay(500);
} // The total delay period is 1000 ms, or 1 second.

* In Arduino, an LED is often connected to pin 13 for "debug" purposes.
* This LED is used as an indicator to make sure that we're able to upload
  programs to the board. It's also a good indicator that your program is running.

```

Figure 2: The source code of the arduino program

## 4 Sender and receiver

For this task we created a ROS project on the raspberry. We then created a python script for publishing and subscribing to a topic within ROS. We then ran the scripts simultaneously in order to achieve a communication between the nodes. The exact steps of this task is described in great detail on the Wiki-ROS webpage <sup>1</sup>.

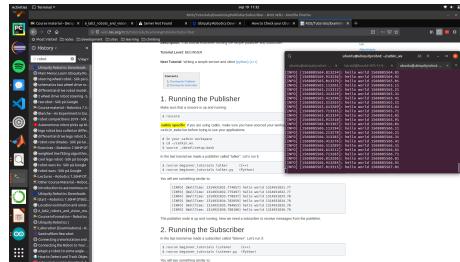


Figure 3: Caption

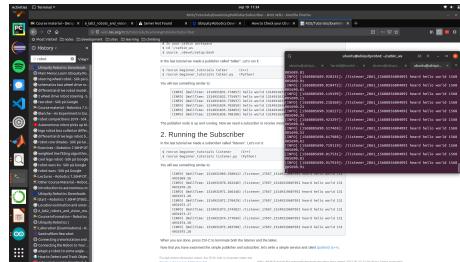


Figure 4: Caption

---

<sup>1</sup><http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29>

## 5 Take a picture

The final task for this lab consisted of using the RPI-camera to take a simple picture. The picture was then moved from the RPI to another computer. The picture is shown below in Figure 5.



Figure 5: The picture taken by the RPI-camera for the final task