*Microprocessor Systems*

**Automated Greenhouse**

**&**

**Monitoring System**

**Alexandru ANTOCE Elvis CARAMETE**

User Requirements 3

Why and how it works 4

System overview 5

Hardware design 6

Repository 8

Results and further work 9

References 12

***User Requirements***

1. The system must provide information relevant to air temperature, air humidity and soil moisture when requested by user.
2. The system should also work fully automated and check if the values from sensors are in the normal parameters.
3. If the values from sensors are not in the normal parameters the system should take control and turn ON the cooler fan.
4. The system has a IP camera which is controlled by the raspberry and it can be seen in the mobile application.
5. The system should have a mobile application which can be connected through sockets with the server which is located on raspberry.
6. Must be easy to use and relatively cheap.

***Green house intelligent control system.***

Green house intelligent control system is designed to protect the plants from more cool and hot weather and additional control system is included to save power by making fans and lights automatically turn on and off with the help of intelligent control system. In this project, the intelligent control system is developed using microcontroller and sensors.

Green house system has a very important use now a day in the agriculture field. Some plants need the specific amount of water for their proper growth and more productivity, therefore farmer should provide them the proper quantity of water.

But it’s difficult for the farmer to get an estimation for quantity of moisture in soil. But in this project moisture sensor is used to provide this facility with a intelligent control system.

***System overview***



**The base subsystem** (which is composed of temperature, humidity and soil moisture monitoring system) encompasses the measurement and data

processing functionalities. The purpose of this subsystem is to acquire

information from the sensors and to process the data obtained from these

sensors.

**Light extension** provides an interface for light related data. The user might

turn light on or off and the subsystem must notify user about light related

changes.

**Cooling Fan** extension provides the user with the possibility to manually

adjust the speed of a cooling fan to decrease the temperature inside

the greenhouse. The cooling fan can also be automatically turned on by the

base subsystem, based on the data provided by the temperature sensor, in

order to decrease the interior temperature of the greenhouse.

**Water Pump** extension provides the user with the possibility to manually turn on an irrigation system, to supply water to the plants. The process of

supplying the plants with the necessary amount of water can be automated

by the base system in the following way: the base system receives soil

moisture percentage from the sensors and controls the water pump

accordingly, until it reaches a soil humidity threshold imposed by the user.

**The Local Data Processing and Storage Subsystem** stores the data

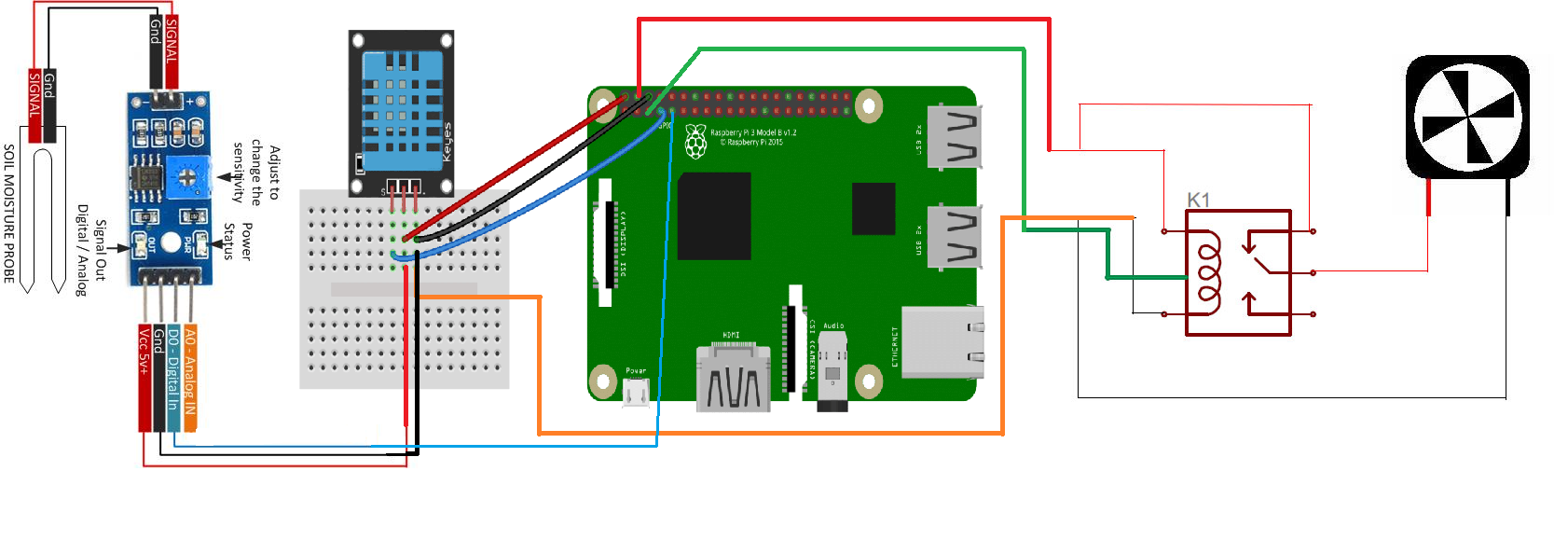
pushed by the **Base Subsystem**. Additionally, it offers a possibility to process

stored information and further send the data to the user.

**Mobile consumer** provides a view representation for the stored data. The

view will be accessed via a specialized application which runs on a mobile device.

***Hardware design:***



**Raspberry Pi 2** provides support for quick prototyping. That makes it a

perfect choice for

quick prototyping but not adequate for real-time applications. We will use the

one-wire interface it has, but also the implicit possibility of communicating with other devices over the Internet.

**DHT-11** is a one-wire enabled sensor. It is a basic, digital-output relative

temperature and humidity sensor.

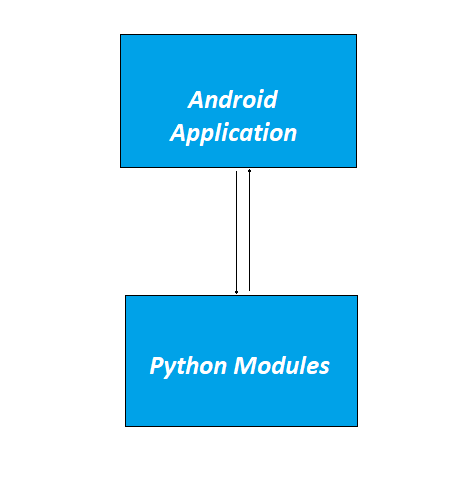
We also use a **generic soil moisture (humidity)** sensor with digital output, which makes it perfect to keep under control the moisture of the soil.

There are 3 relays (5v controlled) which can be used to extend the system for

controlling a water pump, a cooling fan and a light

***Software design:***

The software components and data flow directions. Each of these will be presented in the following subchapters.

******

**Server\_socket.py:**

It retrieves the temperature and humidity tuple from DHT-

11 and it retrieves the soil moisture from the generic sensor.

It controls the cooling and starts the camera. It is connected to the Android Application and listens for commands to turning ON/OFF the cooler fan.

It permanently checks the values from temperatures and display the results.

**Android application:**

It sends data to the python server which is located on

Raspberry.

It can activate the cooling fan and can see the ip-camera through web-sockets.

On the application, we can see in real time results from the sensors, and we can manually activate the cooler fan by pressing a button ON/OFF.

***Repository***

**https://github.com/ElvisCaramete/Microprocessor-Systems**

***Results and further work***

The submitted version of the project offers the following functionalities:

* Reliable reading of the air temperature, air humidity and soil moisture.
* Automated reading from servers and take action in case of irregularities.
* Working IP Camera which can be seen on mobile application.
* Working connection through Mobile Application and Raspberry Server.
* Mobile Application from which we can send massages to Raspberry server to manually activate the cooler fan.

The following list of extensions and/or improvements are planned for the next iteration:

* Implementing also a light sensor and a light source.
* Adding a water pump that can also be activated from Mobile Application

***References***

<https://unity3d.com/learn/tutorials/topics/mobile-touch/mobile-development>

<https://docs.python.org/2/library/socket.html>

<https://learn.adafruit.com/dht/overview>

https://www.modmypi.com/blog/raspberry-pi-plant-pot-moisture-sensor-with-email-notification-tutorial