Politecnico di Milano - Department of Electronics, Information and Bioengineering

NEONATAL INCUBATOR

[Monography]

Student Amin Fathi
ID 853212

Course Embedded System1
Academic Year 2016-2017

Advisor Giuseppe Massari
Professor William Fornaciari

July 20, 2017

Contents

1 Introduction

Premature newborns are small babies that are born before the mother has reached 37 weeks of gestation in other words, they are babies who are born too early. They tend to have underdeveloped body parts that may include the lungs, digestive system, immune system, and skin, according to the American Pregnancy Association.
Since they needed more time in the womb, doctors and nurses in Neonatal Intensive Care Units (NICUs) [1] at hospitals often need to put premature babies inside incubators to control their environment for days, weeks, or even months until they can live on their own in a bassinet or crib.
An incubator also protects premature babies, who are extra vulnerable and are at increased risk of complications, from infection, noise, and light, and it may even provide humidified air to help very premature babies maintain skin integrity.

1.1 Problem statement

Healthcare institutions have varying entry-level requirements for neonatal nurses. Neonatal nurses are registered nurses (RNs) [2], and therefore must have an Ascociate of Science in Nursing (ASN)[3] or Bachelor of Science in Nursing (BSN) [4] degree.
On one hand, in small amount of hospitals, a nurse monitors an incubator, it means the number of neonatal nurses are hired more than enough. In contrast, a nurse can monitor more incubators at the same time on a large screen which a hospital can invest on the other equipment.
On the other hand, a nurse is responsible for more incubators in some of the hospitals. There are some problems in this state of applying a nurse for a group of incubators. First, limitation of hiring many nurses to be responsible for each incubator. Second, due to sensitivity of neonates, it is dangerous to consider only a few nurses responsible for monitoring many incubators. It is difficult for a nurse to check the states of every incubator at the same time and change the actuators.

1.2 Summary of the work

To solve the problem is mentioned prior, an embedded digital system is designed. The system is composed of a data acquisition platform based on Raspberry pi3 and a data communication Lan/Wireless link on TCP/IP protocols. The embedded system is connected to a PC. This embedded system is responsible for monitoring the sensor states.
The main objective of the embedded system is to provide the data acquisition and the communication link with the PC. The variation range of environmental variables in the neonatal incubator are temperature, humidity and velocity of air flow.

2 Design and implementation

2.1 System architecture

According to the layer framework of Internet of Things (IoT), the proposed system is closely divided into four layers. The layer framework of the monitoring system proposed in this application is presented in Figure 1.

The sensing layer will be described as follows. The sensing layer is the lowest layer. The data of temperature, humidity and luminosity is getting from the corresponding sensors. In addition, live video stream is provided by camera.

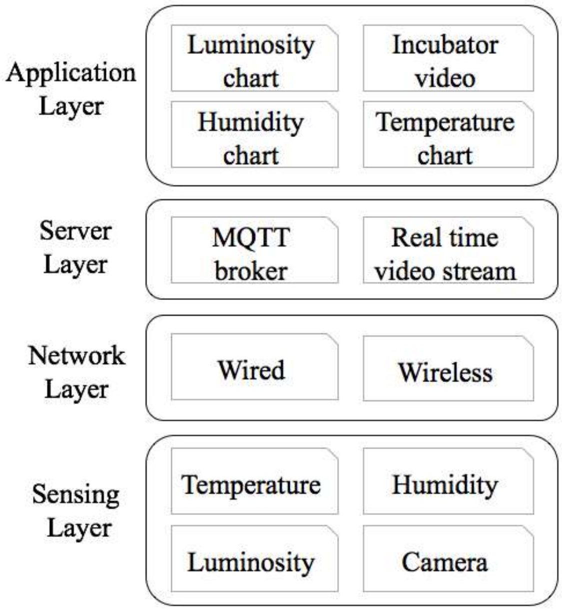

Figure 1: The layer framework of incubator.

The network layer consists of local area network(LAN) which can be connected by wired or wireless configuration. Asymmetric Digital Subscriber Line (ADSL) [5] is used to connect local server to the local network to communicate to the embedded system.
Server layer consists of two components. Message Queue Telemetry Transport (MQTT) [6] protocol is used to transmit string data between server and client. JSON [7], a lightweight data-interchange is used as a data format. The video is sent through TCP/IP protocol. The most widely used programming interfaces for this protocol is sockets.
The application layer provides diverse services to the local network with powerful system interface. Two types of application layer are used on the client and server. Different classes are designed in this layer, such as the humidity/temperature class, luminosity class and camera class. All the data about the baby and incubator's environment is collected by these modules and send to the server to be monitored by nurse and protect a baby from dangerous conditions.

2.2 Design of the hardware

The block diagram of the project is shown in Figure 2. It mainly consists of humidity/temperature sensor, luminosity sensor, pi camera and raspberry pi 3.

The components used in the project have been explained in the following subsections:

2.2.1 Raspberry Pi 3

Raspberry pi 3 is used in this project as a client machine. It has everything we need to work with different sensors. It is a microcomputer based on Quad Core 1.2GHz Broadcom BCM2837 64bit CPU. It has 1GB RAM, wireless/Lan, 4 USB, HDMI, audio port, camera port, display port and 40 input/output pins; These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or

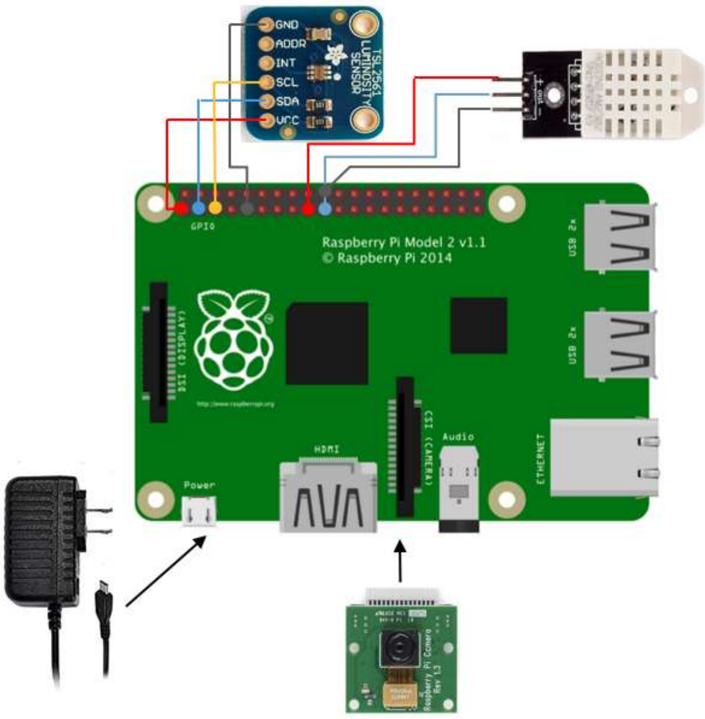

Figure 2: Block diagram of hardware.

off (output). Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins which you should not play with unless you know your stuff!)[8].

2.2.2 Raspberry pi camera

Raspberry pi v2 camera has a Sony 8-megapixel sensor. It supports 1080p30, 720p60 and VGA90 video modes. The camera module can be used to take high-definition video, as well as photographs. It is easy to use for beginners, but has plenty of documents for advanced user. There are a lot of examples online for people using it for time-lapse, slow-motion, and other video cleverness. Using this camera and standard picamera library, it is possible to send live stream to the server through different protocols. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi [9].

2.2.3 Humidity/Temperature sensor

DHT22 digital humidity/temperature sensor is used to monitor the temperature and humidity inside the incubator. The infants have very slow thermal regulation and thermal and humidity are one of the most important factors which affect the preterm. The temperature inside mother's womb is approximately 30°C (100.4°F). Many low-cost sensors have unusual output formats, and in this case, a "Manchester-esque" output that is not SPI, I2C or 1-Wire compatible must be polled continuously by the Pi to decode which is available in python. Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions. Power supply is 3.3-6V DC, accuracy: humidity ±2% RH (Max ±%RH) and temperature $< \pm 0.5$ Celsius, sensing period is 2s. Our code can use any GPIO in, but we will be using #10 pin for our diagram and code [10]

2.2.4 Luminosity sensor

The TSL2561 luminosity sensor is an advanced digital light sensor, ideal for use in a wide range of light situations. Compared to other low-cost light sensors, it is more precise and can be configured for different ranges to detect light ranges from 0.1 to 65500 Lux on the fly. The best part of this sensor is that it contains both infrared and full spectrum diodes! That means you can separately measure infrared, full-spectrum or human-visible light. The sensor has a digital (i2c) interface. You can select one of three addresses so you can have up to three sensors on one board - each with a different i2c address. Its interface is I2C and voltage range is 2.7-3.6v. To wiring up the sensor, we connect the VCC oin to 3.3v power source, connect GND to the ground pin, connect the i2c SCL clock pin to your i2c clock pin (GPIO03), connect the i2c SDA data pin to your i2c data pin (GPIO02) [11].

2.3 Design of the software

The software design of application layer mainly includes the following three parts: application on hardware, application on the PC and server configuration to involve both applications. The process of work is shown in Figure 3. As you can see on the diagram, four distinctive threads are responsible to call "Send Data", "Get Humidity/Temperature", "Get Luminosity" and "Record Camera" functions. Two threads get data from temperature/humidity and luminosity sensors and store them on the class "Data". At the same time, another thread involves camera; whereas, stored data is transmitted by another thread to MQTT broker. The threads are used to process every function separately in parallel due to avoiding crash of application. Following that, some processes are working on application placed on PC by starting GUI. Data are stored on "Data" class which is gotten by class "Get Data"; class "Get Data" is connected to MQQT broker. Every 2 seconds, function "Refresh Chart" is called by a thread to show all the data on diverse chars. Synchronically, live video stream is shown on a player which is configured by the application. PC application is stopped by closing the GUI form. Client diagram describes the functionality of application on raspberry which is shown in Figure 4. In the diagram, every class has specific functions and parameters does a significant part of whole application.

- Class "main": IPAddress is a variable which is point to server IP. It is stored in file "config". Every function seen in the diagram initiate a thread. Moreover, every thread calls another class.
- Class "sensor": this is a standard library of python which communicates to the DTH22 sensor directly.
- Class "TSL2561": this is another standard library which is coded to ease development of higher application level.
- Class "Data": only temporal data is stored inside this class. All the data is removed by closing the application.

Server diagram illustrates the structure of application on PC which is presented in Figure 5.

- Class "clsMain": is responsible to the full functionality of the application. Three threads are used to instantiate other classes which is described on the picture.
- Class "subscriber": a MQTT subscriber is used in this class to communicate with the MQQT broker and receive data.
- Class "Data": "quLength" is a variable store maximum packet of data; but, the data is pushed on other variables. There are four "inc" variables, every "inc" store data from a incubator. We are monitoring four incubators in this project.

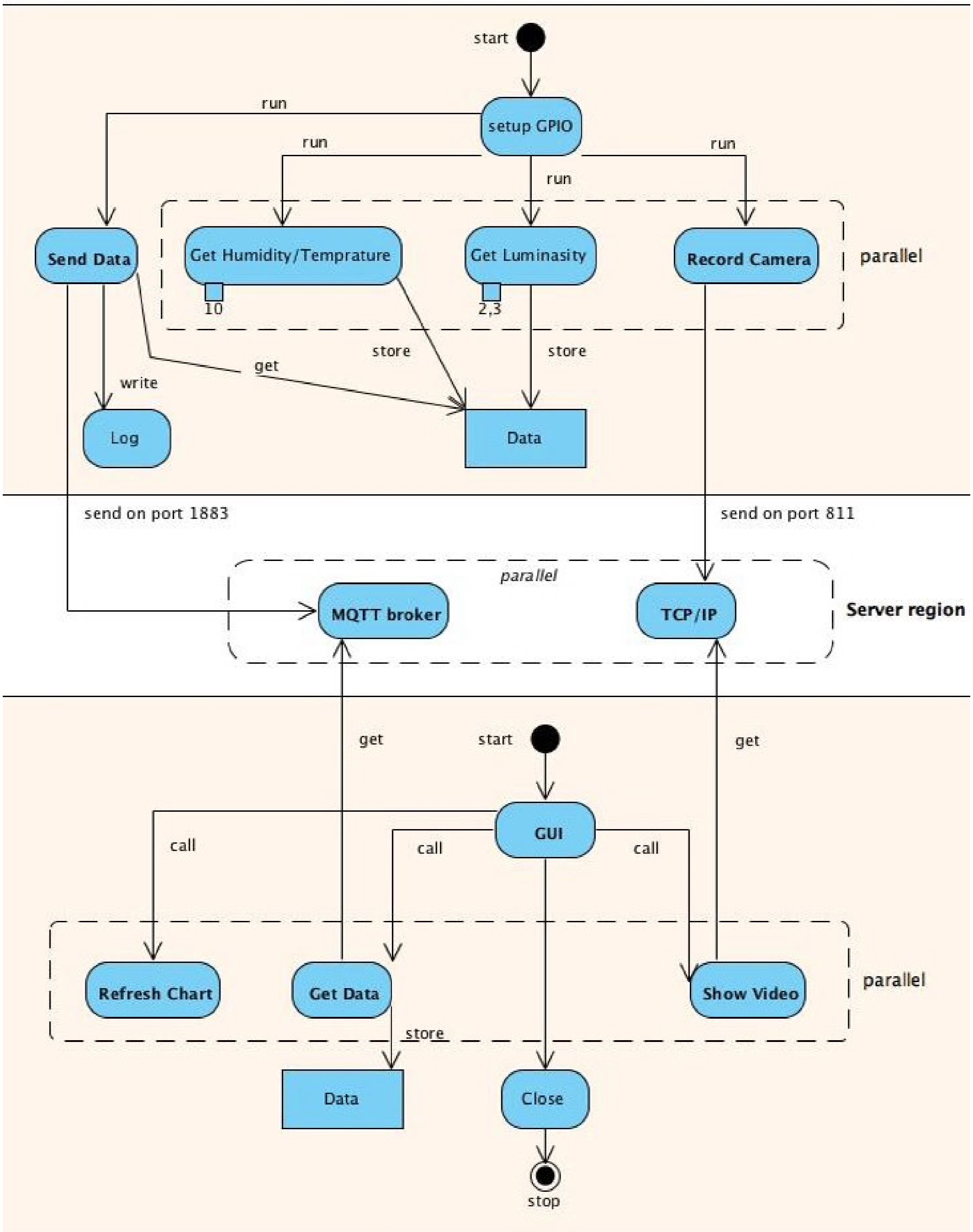

Figure 3: Software work flow.

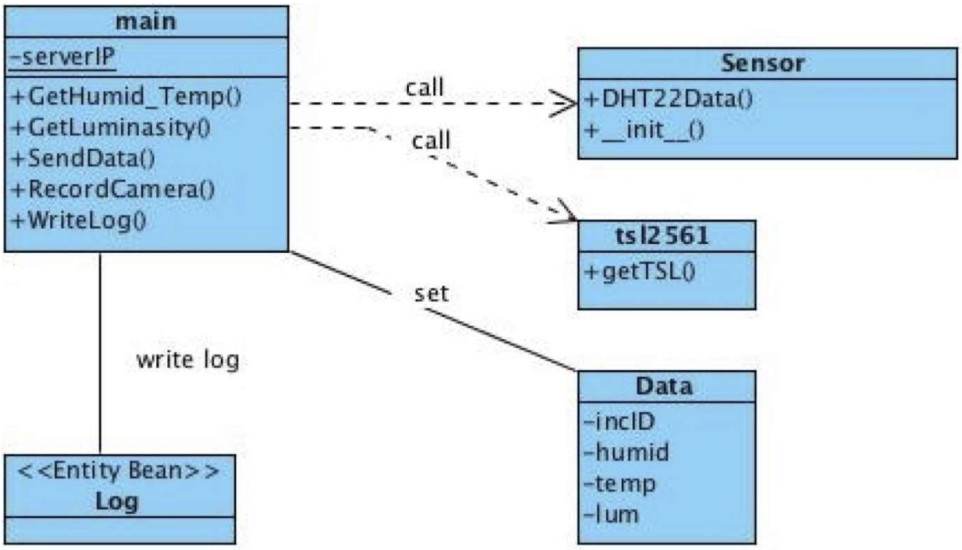

Figure 4: Client class diagram.

- Class "UI_Form": draws objects on the form.

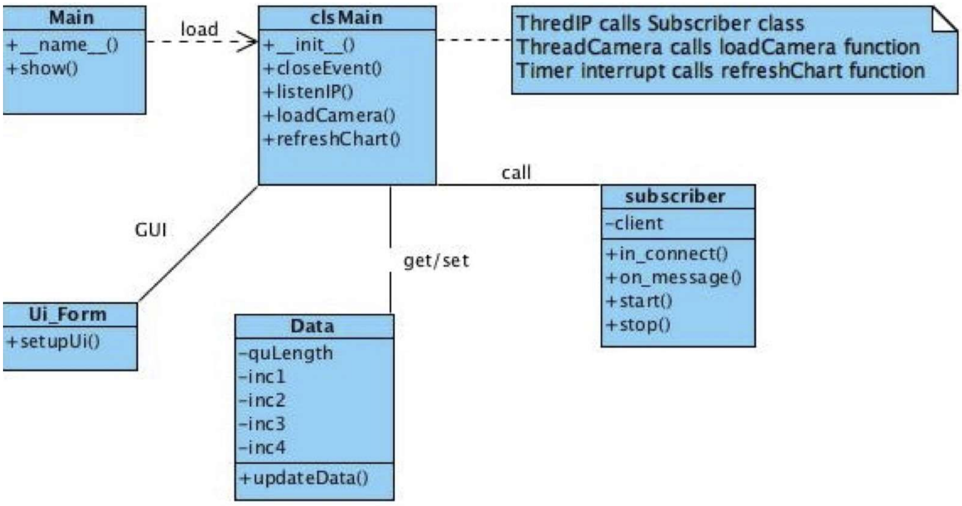

Figure 5: Server class diagram.

3 Experimental evolution

3.1 Experimental setup

To run whole project, there are different steps must be considered. In the following we will focus on the client (raspberry pi).

1. Download and install RASPBIAN OS on raspberry [12]. You can also use PiBakery [13] free application.
2. Enable camera [14] and I2C [15] on raspberry by using "`raspi_config`" command.
3. Install "paho-mqtt" python library [16].
4. Install "smbus" python library [17].
5. Set IP server on the "config" file from client code.

The second step is the configuration of server as a bridge between hardware application and monitoring application.

1. Install and run "mosquitto mqtt broker" on the server [18].

Finally, monitoring application's requirements must be installed.

2. Install "pyside" python library [19].
3. Install "pyqtgraph" python library [20].
4. Install "paho-mqtt" python library [16].

To get the best result, we should do every step sequentially. Strongly recommend you to run monitoring application on the server where MQTT broker is installed; otherwise, IP on monitoring application should be set to IP server.
To simulate a baby incubator which send data from embedded system to server Client List of hardware parts: 1- Raspberry pi 3 2- DHT22 humidity/temprature sensor 3- TSL2561 digital luminasity sensor 4- Pi camera Camera and I2C should be enabled on raspberry. List of libraries are used in client application: 1- smbus 2- atexit Server List of libraries are used in client application: 1- pyside 2- mplayer 3- pyqtgraph 4- subprocess.

3.2 Results

Through design of hardware and software for the monitoring multiple neonatal incubators, the information of the environment in the incubator can be obtained by the sensors.
Based on analyses above, the curve of temperature, humidity and luminosity in 15 seconds are shown in Figure 6. We can clearly find that the temperature is held in the range of 25°C to 26°C under the air condition in the normal room not in incubator. This range is not relevant to premature babies. The relative humidity is shown between 61% to 63%. The light trend is shown from 120 lux to 160

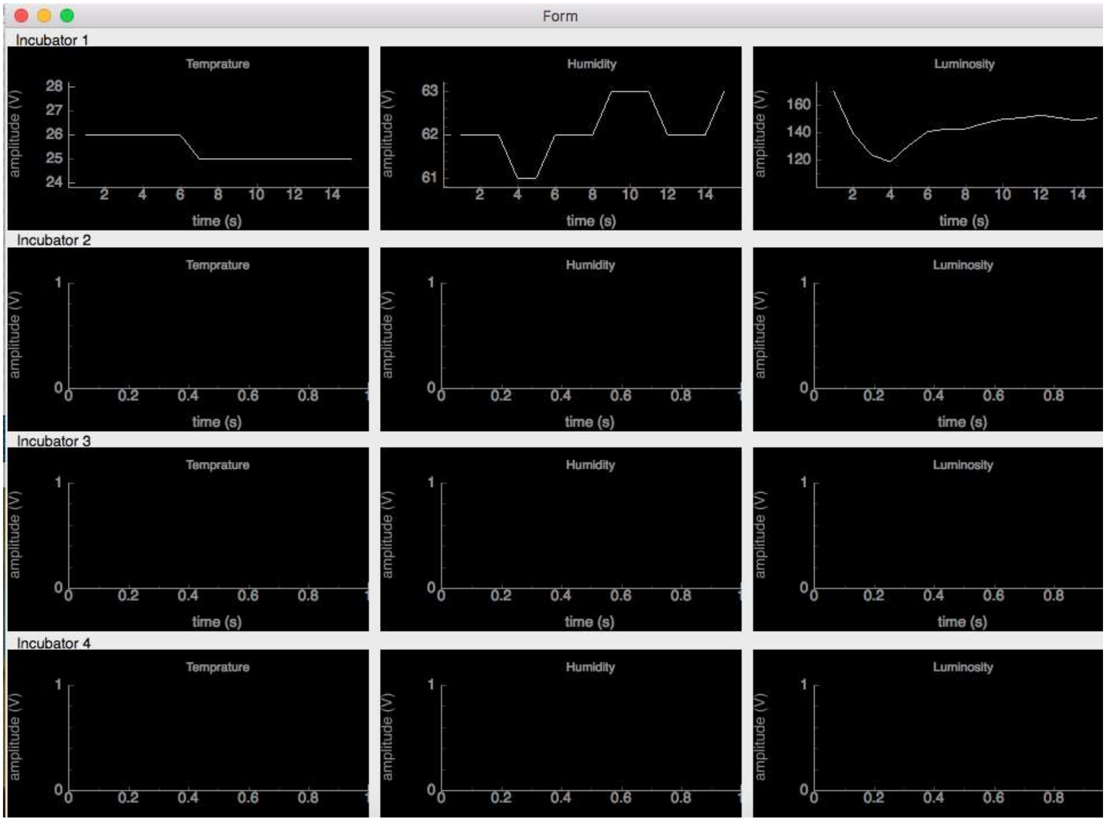

Figure 6: Result of first incubator.The other incubators are disabled.

Figure 7: Picture of new born baby taken by camera.

lux; the luminosity value should be converted to ideal value which is understandable by the user. The movement of baby is shown in Figure 7.

4 Conclusions and Future Works

The objective of this project is to design and develop a monitoring application which shows the result of at least four incubators on a big screen. Hospitals do not need to allocate one nurse to one incubator, now one nurse for four incubators. The experimental results show that this system can provide a real-time and reliable management for the infant incubator system.
Future scope:
As we have used four sensors (humidity, temperature, luminosity, camera). However, it can be improved by future using other sensors like Oxygen, Pulse etc. In addition, another interesting development of such system is monitoring the results on the web platform through the Internet. Moreover, design a control system to inspire oxygen to the incubator box.

References

[1] https://en.wikipedia.org/wiki/neonatal_intensive_care_unit.

[2] https://en.wikipedia.org/wiki/registered_nurse.

[3] https://en.wikipedia.org/wiki/associate_of_science_in_nursing.

[4] https://en.wikipedia.org/wiki/bachelor_of_science_in_nursing.

[5] https://en.wikipedia.org/wiki/asymmetric_digital_subscriber_line.

[6] http://mqtt.org.

[7] http://www.json.or.

[8] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[9] https://www.raspberrypi.org/products/camera-module-v2/.

[10] https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/overview.

[11] https://www.adafruit.com/product/439.

[12] https://www.raspberrypi.org/downloads/raspbian/.

[13] http://www.pibakery.org.

[14] https://thepihut.com/blogs/raspberry-pi-tutorials/16021420-how-to-install-use-the-raspberry-pi-camera.

[15] https://learn.sparkfun.com/tutorials/raspberry-pi-spi-and-i2c-tutorial.

[16] https://pypi.python.org/pypi/paho-mqtt/1.1.

[17] http://skpang.co.uk/blog/archives/575.

[18] http://www.xappsoftware.com/wordpress/2014/10/30/install-mosquitto-on-mac-os-x/.

[19] https://pypi.python.org/pypi/pyside/1.2.4.

[20] https://stackoverflow.com/questions/39249733/installing-pyqtgraph-on-ubuntu.