Smart home object detection (GIAN System)

Grigoryan Vahe
ID: 1600567
Supervisor: Thakur Manoj
2nd assessor: Reed Martin
Degree course: Computer Systems Engineering

1.Abstract

I present General Interactive automation network (GIAN) System project. In this report I will describe the project goal, background reading and project planning. To provide evidence that this project is possible to do with available technologies, I will build parts of the project in small scale to test them, and if necessary to find better solutions to existing problems.

2.Project goal

The goal is to build a system that can recognize hand gestures and operate units (Cars, furnitures) with those gestures. The idea is to have a system like this in a smart home for user to operate the furniture with simple gestures. It will be very costly and challenging to make the full-scale system at this stage, so a small prototype must be built instead. To make sure that the developed code can be used in real envronments It is important to make the system scalable so that more units and communication protocols can be added in the future.

Since it is an interactive system the whole thing must run in real time at reasonable speeds and the network must support multiple protocols or must be a combination of different networks. Finally, the system must work with large objects in large environments without significant changes to its structure.

When thinking about previously mentioned features it becomes clear that the project can get very complex and confusing if it is not done properly, therefore while I do the research I should make sure that the technologies used for building the system are chosen wisely and if necessary are simplified to do only the tasks that the system needs.

3.Background reading

Before starting the research, I broke down the project into multiple parts to be researched and developed, which then should be combined into the final thing. The parts are hand gesture detection, “unit” position detection, “unit” control, LED matrix control, product enclosure (case) and programming languages. I keep a logbook for this project which can be accessed here [1].

4.Hand gesture detection

Internet is full of papers describing amazing hand gesture detection technologies. Most of them require multiple sensors and markers to work, which means that the user must have markers on their hand for system to detect the gesture. This method is not good, because it limits where the system can be used, also it requires a lot of code to be written to hardcode marker positions that correspond to gestures. For example, this paper [2] uses “code book algorithm” to process camera input by separating users’ hand from the background for detecting the number of fingers shown in the input. The problem here is that a “haarcascade” for each gesture must be made for OpenCV to detect the gesture. This paper [3] uses “fuzzy clustering” technique for activity recognition. The technique presented in the paper can be adapted to work for my project, but it has one problem, it works with “Microsoft Kinect”, which is a camera unit with additional sensors. It is a problem because for this project I want to use only cameras as sensors, because it will enable the system to work in different environments. The researcher of this paper [4] has also preferred to use camera over secondary input device (Data-Glove) for sign recognition, and they explain the decision with the following “the devices are quite expensive and bring much cumbersome experience to the users”. For those reasons I decided to continue my research focusing on neural networks and machine learning papers that feature only cameras as input devices, because these techniques do not need user coded gestures and additional sensors to work. The following paper [5] describes a software that attempts to control mouse cursor by users’ hand. This relates to my project, because I am attempting to control LED matrix by hand gestures. From the paper it becomes clear that I need some sort of “classification” and “feature extraction” algorithms to detect the hand and track its location. I found this “pose detection” paper [6] from Cornell University library, which is very interesting, because the system uses only camera footage to determine persons pose, but to make it work

a lot of data must be collected and annotated with the correct poses, which is a lot of work for one person. Fortunately, I found this other paper [7] that also uses only camera footage and it comes with built in “classifiers” and “feature extractors”, but instead of annotating the whole pose it works with labels, which are telling what the network is seeing, this approach is less time consuming, thus manageable by one person. The paper describes YOLO (you only look once) method. This method is better at object detection than other object detection techniques, because it scans the footage only once and can find multiple object at once. Other techniques such as R-CNN or fast R-CNN take more time and resources to do the same thing, but they have the advantage of being more accurate. Since my system is supposed to work in real time YOLO is much preferred option for now.

I am going to use one camera to detect users hand gestures and another camera to detect the “unit” position. Good quality high definition cameras are expensive, therefore to keep the cost down I might use my smartphone as the second camera.

The hardest bit for using YOLO is the installation and data collection processes. To get the YOLO up and running multiple external libraries and drivers must be installed, including GPU drivers and libraries to enhance the performance of the software. Fortunately, the installation processes of TensorFlow [8] and YOLO [9] are describer ferly well on their official web sites.

The data collection means collecting images of people pointing to camera and that of showing “OK” sign, which then must be annotated. Annotating images means creating an xml file for each image that holds data of where the hand is in the image and what gesture it is showing. To create those xml files, I will use a free software “LabelImg” [10]. This software has a graphical user interface, which makes it easier to create xml files by simply highlighting the needed part in the image. After collecting and annotating some images, a convolutional neural network can be trained using the method described in the YOLO website.
Finally, I will use the trained model to detect gestures from the camera footage and will display the results using OpenCV.

5.“Unit” position detection and control

For the “unit” I can use a Lajos NXT to build a robot that I can control remotely. For detecting the location of the robot, I can make another YOLO model that recognizes the robot on a grid from a camera input and extracts the coordinates of the robot from the frame. I can also use DEAD Reckoning localization with the robots’ on board odometry, the advantage of this technique is the fact that it is less computationally intensive to perform and can be done by the robot, however it will not work if robots’ position is needed when it is stationary, the opposite is true with YOLO model, and for this reason I decided to use both, so when robot is stationary I will use YOLO and will use odometry when the robot is moving to find its position. Two methods can be used for controlling the robot. Data can be sent in real time, via Bluetooth, for robot to go forwards, backwards, right and left, or coordinates can be sent, based on, which robot will do calculation to figure out where it needs to go from its current position. The second method is preferable, because it does not require robot tracking, so I will use the YOLO model only for detecting the robots’ initial position and then robot will do the rest, allowing more CPU and GPU time for other tasks. This can be useful if in the future more robots are added to the system. In this paper [11] a robot is controlled with gestures as well, but it is not using the YOLO for gesture detection and the robot is controlled by sending directions, instead of locations. From the paper, it is clear that sending directions while the robot is moving introduces delays, because of communication, which leads to inaccurate control. It is Interesting to note that the researchers’ object detection method had 90% accuracy, which is impressive and proves that it is possible to achieve high levels of

accuracy with machine learning techniques.

6.LED matrix control

There is a need for making a LED matrix, because user needs to have some sort of feedback from the system to help them intuitively use it. In this case the LEDs of the matrix will show where the hand of the user is pointing. There are many ways for making such matrices, in this paper [12] I found very detailed descriptions of power LED drivers, these methods can apply to my project, even though I am using regular LEDs instead of power once. The implementation of those drivers can get very complex when using large number of LEDs, so I continued my research and found many other methods. Charlieplexing, address registers with controller and multiplexing just to name a few.

The matrix is going to be on a 50x50 cm case, so my plan is to have one led per 4 cm, which means that I have to build 12x12 matrix, thus 144 LEDs must be addressed individually. With charlieplexing many LEDs can be addressed with a few controller pins, but it is hard to implement on a PCB and if one of the LEDs stopes working it will cause other LEDs behave incorrectly, for those reasons I will use the next best method, which is multiplexing. This method needs more controller pins, but it is easier to implement, and faulty LED does not affect the other LEDs. To make the job easy I will make four 6x6 LED matrices, where each matrix will have an Arduino pro mini to control the LEDs and communicate with other matrices using I2C communication protocol. 12 pins from the Arduino will be used for controlling the 6x6 matrix (36 LEDs), which means I don't need to use external address registers because the controller has enough pins, however I must use an external power source because Arduino does not have enough power to control 36 LEDs at the same time. The following Multisim project [P1] shows the circuit design for 36 LEDs.

7.Project case

Some sort of enclosure is needed for keeping all the components of the project together and the robot needs to have a space where it can be manipulated. For those reasons a case must be built. The case can be made of wood and it will probably have square shape on bottom and will have a spot on top where cameras can be attached. The LED matrix will be placed on the bottom of the case and it can be cover with acrylic glass to let the light through, and finally the robot will be operated on top of the glass.

8.Programming languages

The neural network that detects objects is the crucial part of the system, so the language that has the most flexibility and support, when working with machine learning frameworks, will dominate the system in terms of programming languages. Python with its simple syntax and powerful 3rd party libraries is the choice for this system. Python is good for machine learning, but when it comes to low level programming such as controlling LEDs the programming language C is much better, and it works with Arduino IDE, which I am going to use to program Arduino boards. Java will be used for programming the robot because it has good support libraries for Lejos nxt, which will make the programming process a lot easier. Java might also be used for making an Android app to use the smartphones camera for detecting the position of the robot when it is stationary.

9.Testing

For object detection I decided to use the YOLO method, but before I can do that, I need to install drivers for GPU, many 3rd party libraries to support TensorFlow and Darknet, and finally use the provided trained model to test the installation. (More about the installation in the main report).

After completing the above steps, I made a simple python script that takes a footage from a web cam passes it to YOLO which then does the prediction on the laptops GPU, and finally it displays the results.

The point of doing this was to calculate the time it takes for detection to happen on my laptop. The results were a bit disappointing but acceptable. The program run at 10 fps, after optimizing the code with multiple threads the frame rate went up to around 15 fps.

At this point CPU was usually waiting for the GPU to respond, which means for further performance gain the GPU had to be replaced with better one, but I think 15 fps will do for this project.

The next things that needed to be tested were the Multiplexing with LEDs and I2C communication. To test them I made 3x3 LED matrix and used one Arduino board to control the matrix and another one to choose which led should lit up. The communication between the Arduinos was established via I2C. The results were impressive, almost instant communication was achieved, but to make the matrix work I had to add NPN transistors in to the mix. [P2] (More about the implementation in the main report).

10.Project Planning

The system is designed in a way that different parts of it can be developed independently, meaning I can work on different parts of the project even if I haven't finished working on the others. The benefit of this is the ability to switch tasks when one becomes exhausting, also sometimes a problem in one part of the system can be solved in other part of it, so the development process does not stop when I come across with a difficult problem. This agile approach was carried out during the summer research, the development progress can be found here [13]. I am planning to work the same way to develop the whole system, and I believe I can make good assumption about how long it would take to develop different parts of the project based on my summer work.

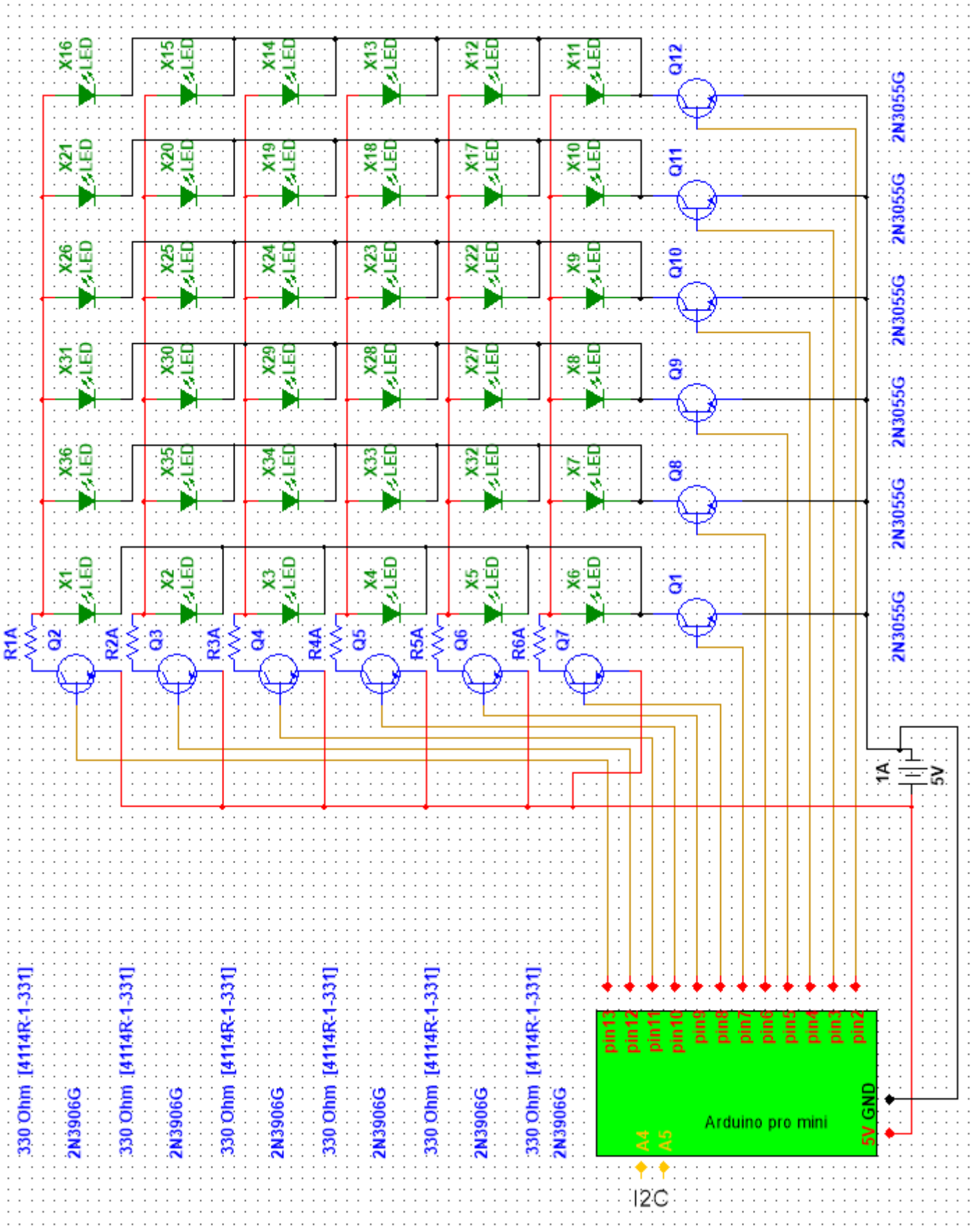
The following [P4] shows the development process and the relationship between the different parts of the project.
On the right side of the image you can see the task, the number of days for completing the task and number of total hours that will be spent on each task for completing it. On the left side you can find the project flow (meaning the relationship between the tasks), the resource and a percentage showing how busy the resource is going to be for each task. For example, "14%" means the resource will spend that much of available days for working on the task.

The following table [P5] shows how much of the task is completed and how many days it is going to last.

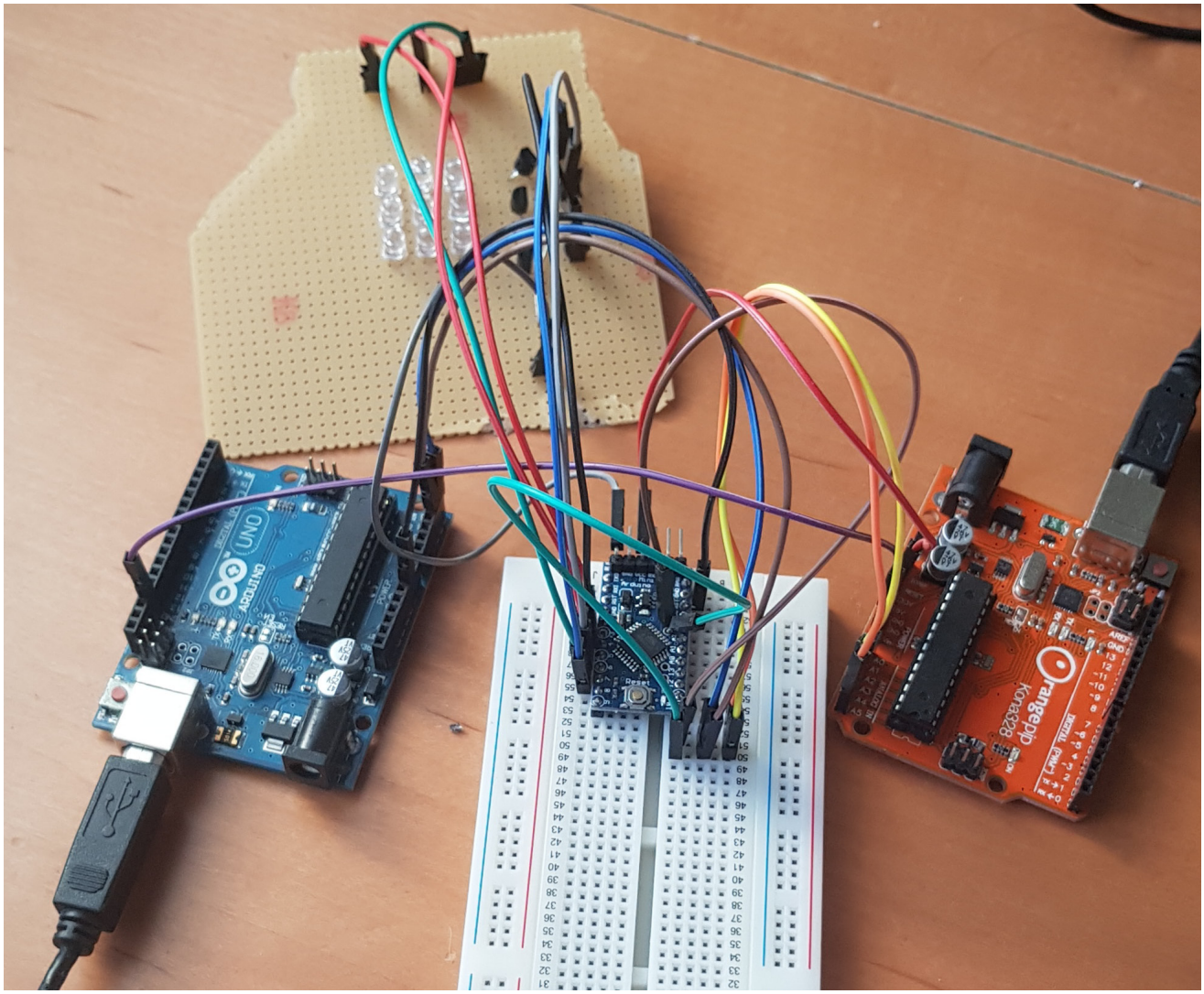
This table [P3] show the milestones that should be achieved at the specified dates, this information can also be found on project flow diagram [P4].

From that table we can see that two milestones will be achieved before the interim oral examination, first one is the finished LED matrix and second one is that the matrix we be controlled with gestures.

The project will be managed with the JIRA [14] instead of Taiga server, which was used over the summer, also all the files and papers mentioned in this report, as well as the project files can be found in this GitLab repository [15].



P1. Initial LED matrix circuit design



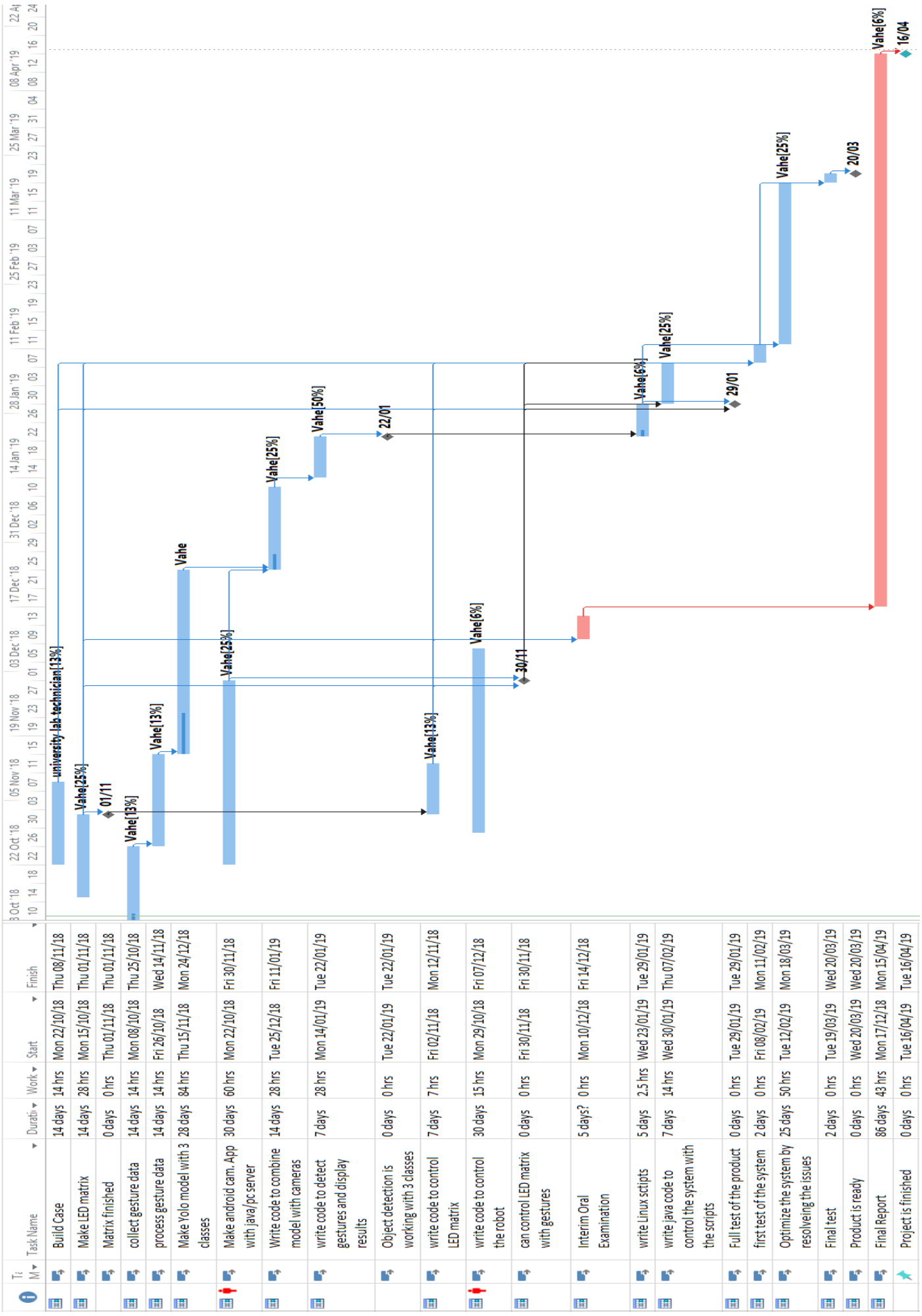
P2. Prototype for testing LED multiplexing and I2C communication

P3. The Mailstones

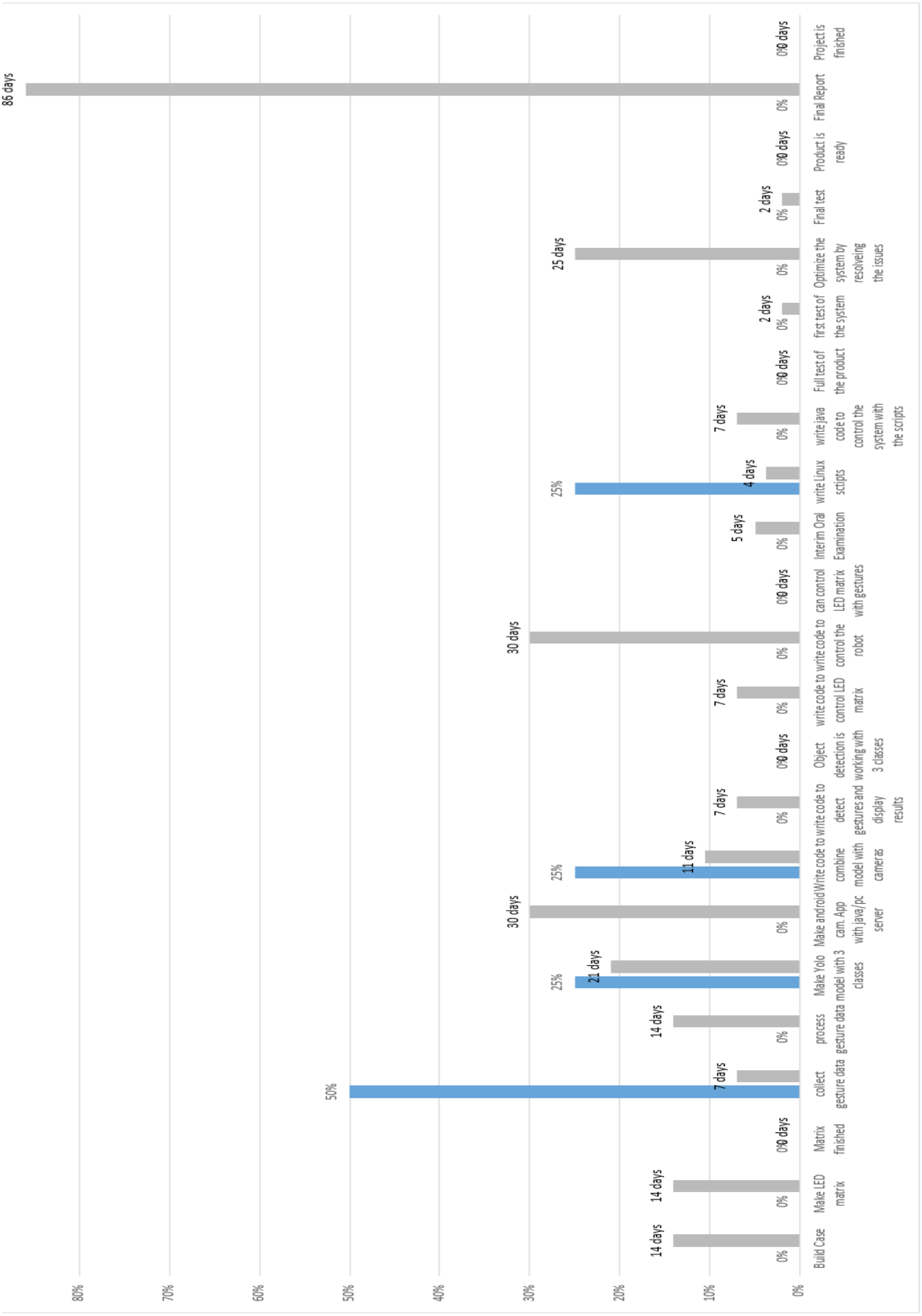
MILESTONES DUE

Milestones that are coming soon.

Name	Finish	Late Finish
Matrix finished	Thu 01/11/18	Thu 29/11/18
Object detection is working with 3 classes	Tue 22/01/19	Thu 24/01/19
can control LED matrix with gestures	Fri 30/11/18	Fri 05/04/19
Full test of the product	Tue 29/01/19	Tue 16/04/19
Product is ready	Wed 20/03/19	Tue 16/04/19
Project is finished	Tue 16/04/19	Tue 16/04/19



P4. Time estimation for different tasks of the project



P5. Total days available and % of the work completed per task

References

[1] V. Grigoryan, “Logbook”, OneNote, 2018. [Online]. Available: http://bit.ly/2yf5Fdn. [Accessed: 16- Oct- 2018].

[2] M. Srinivasa, “Implementation of Real Time Hand Gesture Recognition”, International Journal of Innovative Research in Computer and Communication Engineering, 2015.

[3] T. Udhayakumar and S. Anandha Saravanan, “Activity Recognition from Video in Day or Night Using Fuzzy Clustering Techniques”, International Journal of Innovative Research in Computer and Communication Engineering, 2014.

[4] P. Chaudhary and H. Singh Ryait, “Neural Network Based Static Sign Gesture Recognition System”, International Journal of Innovative Research in Computer and Communication Engineering, 2014.

[5] D. Suresh, P. Subash and F. Noorullah Khan, “Computer Intraction Based on Hand Gestures”, International Journal of Innovative Research in Computer and Communication Engineering, 2015.

[6] R. Alp Guler, N. Neverova and I. Kokkinos, “DensePose: Dense Human Pose Estimation In The Wild”, 2018.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[8] ”Install TensorFlow with pip | TensorFlow”, TensorFlow, 2018. [Online]. Available: https://www.tensorflow.org/install/pip. [Accessed: 10- Sep- 2018].

[9] J. Redmon, “YOLO: Real-Time Object Detection”, Pjreddie.com, 2018. [Online]. Available: https://pjreddie.com/darknet/yolo. [Accessed: 19- Sep- 2018].

[10] Tzutalin, “tzutalin/labelImg”, GitHub, 2018. [Online]. Available: https://github.com/tzutalin/labelImg. [Accessed: 16- Sep- 2018].

[11] M. Dhabale and A. Kamune, “Gesture Identification Based Remote Controlled Robot”, International Journal of Innovative Research in Computer and Communication Engineering, 2014.

[12] R. AKKAYA and Y. GÜRBÜZ, “Design, Implementation and Comparison of Flyback Type Power LED Drivers with External and Internal PFC in Terms of Harmonics and Power Factor”, International Journal of Innovative Research in Computer and Communication Engineering, 2014.

[13] V. Grigoryan, “Taiga”, Cseetaiga.essex.ac.uk, 2018. [Online]. Available: https://cseetaiga.essex.ac.uk/project/vg16661-object-detection-for-autonomous-driving. [Accessed: 16- Oct- 2018].

[14] V. Grigoryan, “Log in - CSEE Jira”, Cseejira.essex.ac.uk, 2018. [Online]. Available: https://cseejira.essex.ac.uk/browse/PC-27. [Accessed: 16- Oct- 2018].

[15] V. Grigoryan, “general-intereractive-automation-network”, GitLab, 2018. [Online]. Available: https://cseegit.essex.ac.uk/vg16661/general-intereractive-automation-network. [Accessed: 16- Oct- 2018].