

MIDDLESEX PLATFORM MEASURING SYSTEM

CCE 4999 PROJECT REPORT

MENG PROJECT 2018

SHIRWA HAJI M0044139

ANCILA ULUNGI M00517972

24TH April 2018

LIST OF CONTENTS

Introduction

This is a report which discusses a group project which was conducted for the past 24 weeks in this academic year which was to create a system that measures the accuracy of a 6DOF platform. The report will discuss the main purpose of the project, how the project was implemented, the outcome of the project and the testing of the system as well as what would have been done differently for a better delivery.

Purpose of the project

The main purpose of this project was to create a system that measures the accuracy of the 6DOF platform which is a chair that is used for virtualization of different simulations like, roller coaster or plane flight assimilations. This chair is made up of six actuators which move up and down and which are of different lengths that can be measured to see if they move accordingly to the commanded data. The chair has also a platform which moves in three angular degrees and so the system will measure the accuracy of the angles moved. The following is a picture of the 6DOF chair that the system will measure focusing on its accuracy.

A close up of a bicycle

Description generated with high confidence

Fig 1: 6DOF Platform Chair

As we can see from the picture there are six actuators which hold the platform which the chair is mounted on at the end of each actuator distance sensors are attached to it so that they can measure the change in length of the actuator as the chair moves. The project intends to create a system called the Middlesex Platform Measuring System which will measure system errors of the 6DOF platform chair.

Functional Overview

The system created in this project has specific functions that it needs to perform for perfect results of the system. These functions in detail can be found in the functional specification document for the Middlesex Platform Measuring System but an overview of the function is divided into two different parts the Sensor Server and the Monitor functionalities.

The Sensor Server:

This is involved in capturing the actual motion of the platform which will read the actual positioning and movement of the platform in real-time using distance and angular sensors. These distance and angular sensors measure the lengths of the actuators and roll, pitch and yaw values for the angles. Once these values have been read they need to be sent to the monitor for comparison with the commanded values and the format used for the sending the data consists of a header followed by six actuator lengths represented in Asci digits and three angular measurements represented in floating point numbers.

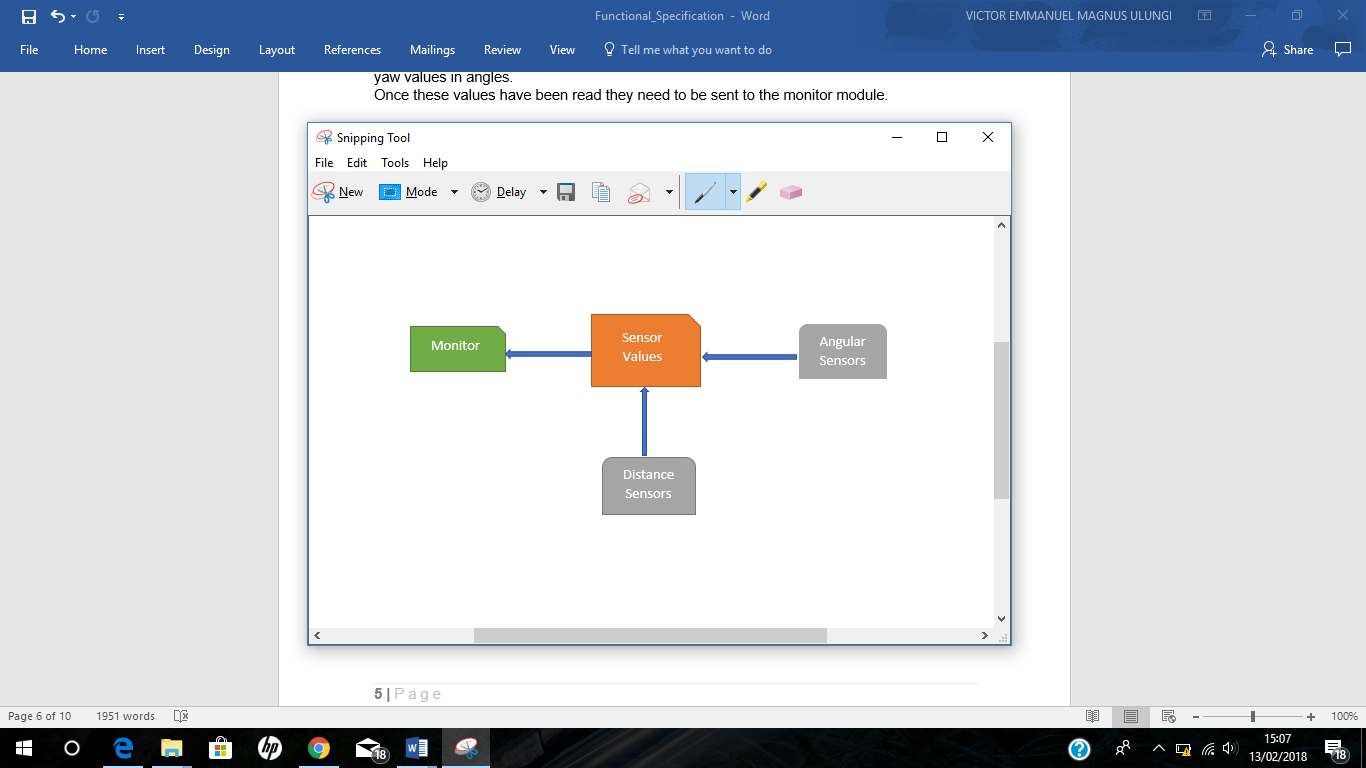


Fig 2: Sensor Server Block Diagram

For system overview see functional specification for details.

The Monitor:

This part is involved with sending commanded data to the middleware, receiving actuator lengths from the middleware, receiving sensor data from the Sensor Server, comparing the two sets of data and storing and displaying the error. The saved data will be saved in a csv extension file. The Monitor expects to receive the data from the Sensor Server in the same format that the Sensor Server send the data.

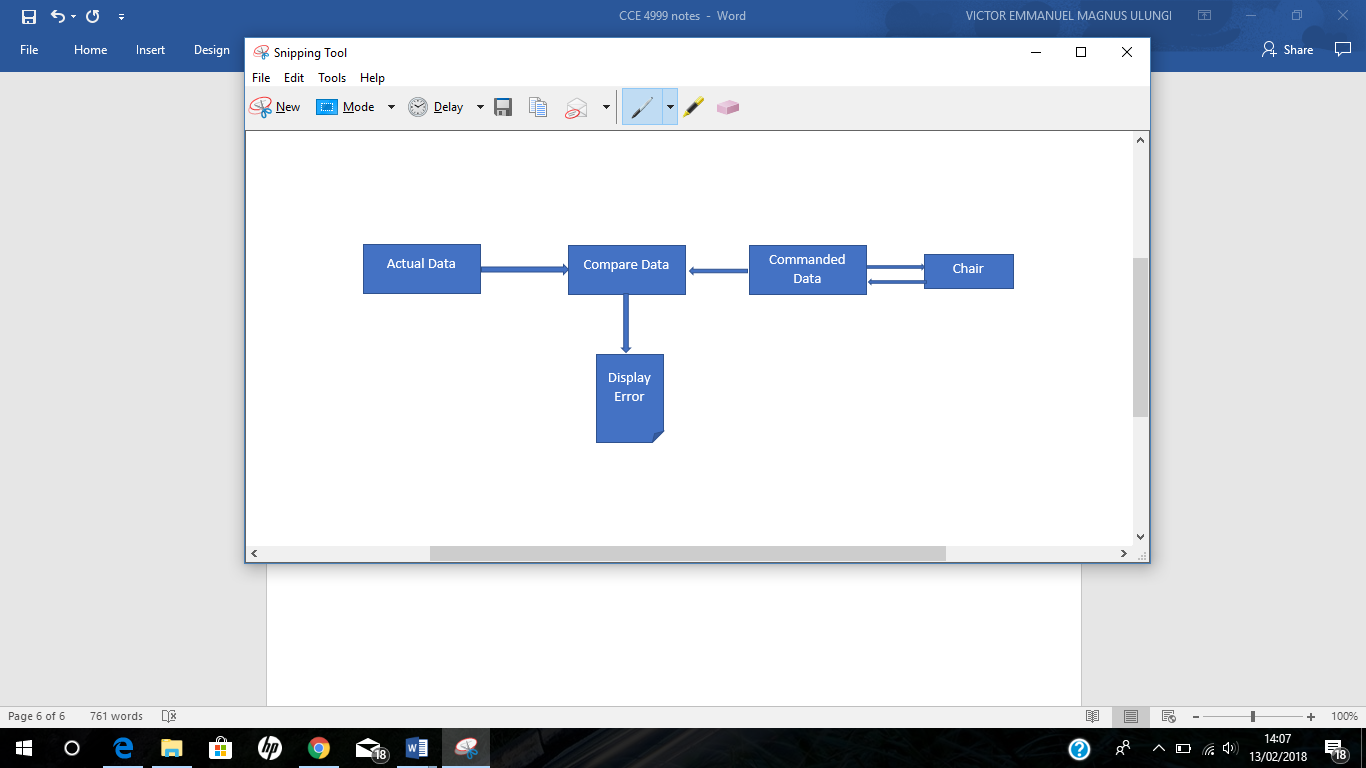


Fig 3: Monitor Block Diagram

Design Overview

The system design consists of two parts which is the software design and the hardware design to make up the complete system design.

The Hardware design:

The hardware used in this project consists of distance sensors, Arduino Mega, Arduino Uno and an angle monitor module called jy901.

The distance sensors used to measure the lengths of 6 actuators are Maxbotic sonar sensor MB1043 HRLV-MaxSonarEZ4. These sensors are attached at the end of each actuator and are capable of reading 30cm to 5000cm with accurate precision up to the nearest 1mm on their serial pin.

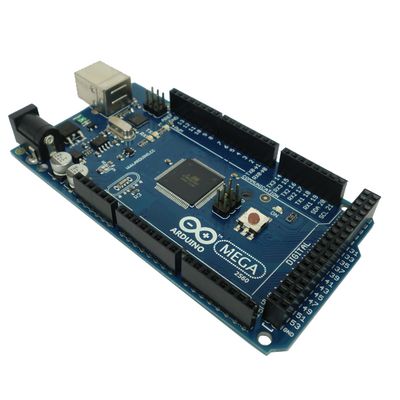
The Arduino Mega2560 is a microcontroller board based on ATmega2560, it was chosen because it has many number of pins and it can handle several sensors at the same time. To be able to connect all the sensors to the board a distance sensor adapter board is used. This board was created to connect all the six sensors to the SoftwareSerial ports in an easier manner.

The Arduino Uno is a microcontroller board based on the ATmega328p. Within the Uno there is an angle monitor hardware module called jy901 which is used to measure the angles.

To be able to send data from the Arduino Mega to the Uno we connect the two boards using jumper wires.

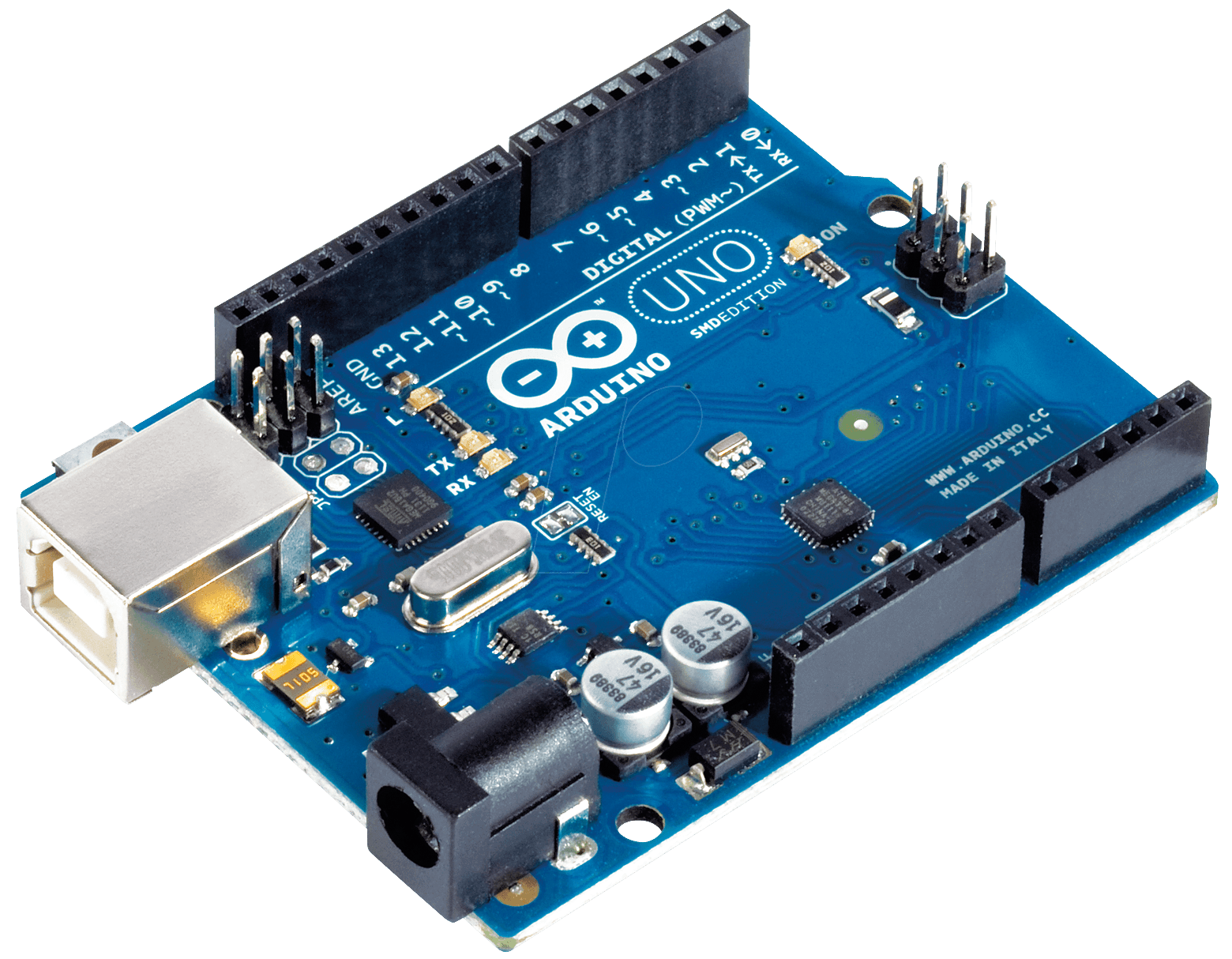
Below is a description of the hardware used

*Arduino Mega2560*

The Mega 2560 - a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

See design document for further details.

*Arduino Uno*



Arduino/Genuino Uno - a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

See design document for further details.

*MB1043 Sonar Sensors*



The HRLV-MaxSonar-EZ4 is the narrowest beam width sensor that is also the least sensitive to side objects offered in the HRLV-MaxSonar-EZsensor line. The HRLV-MaxSonar-EZ4 is an excellent choice when only larger objects need to be detected.

For more details see the design document.

*Technical Specs*

|  |  |
| --- | --- |
| Reading Rate | 10 Hz |
| Current Consumption | 3.1 mA |
| Operating Voltage | 2.5V - 5.5V |
| Maximum Range | 5000 mm (195 “ ) |
| Resolution | 1 mm |
| Operating Temperature | -15ºC to +65ºC |
| Sensor | 42 kHz |

The Software design:

Two systems are involved within the software design which are Sensor Server and Monitor module. The sensor server involves two modules which are distance sensors and angle server.

The distance sensor involves reading six distance sensor values using SoftwareSerial. After the distance sensor reads the actuator lengths sends the data to the Angle server but if the distance sensor does not read any data then it sends a zero-value indicating that there is no data coming from the distance sensor.

The Angle server reads the 3 angular values using jy901 library which provides a method that reads the sensor angle data. It also receives the distance sensor data and creates a message format including both data the distance data and the angle data and sends it to the Monitor in the format as mentioned in the functional specification and will send the data as soon as it is available.

Implementation Overview

To implement the project the hardware and software had to be prepared as described in the design overview. To start with Arduino sketches had to be created and uploaded on to the boards which are used to retrieve data from the sensors. Two sketches were involved where one read only actuator lengths via the mega board and the other sketch measured the angles and at the same time retrieved the distance sensor data from the mega board to the Uno and send the combined data to the monitor.

The other part was to create a python script which sent commanded data to the middleware and received the actual sensor data from the sensor server and compared the sensor data and the commanded data then stores and displays the difference.

Positioning of the hardware:

The Uno and the Arduino Mega board are placed on one board and that board will be placed on the chair and the sensors will be connected to the distance adaptor board using jumper wires.

The six sonar sensors will be mounted on the fix ends of the actuators. An actuator is like an arm with one fix end and the other one moving. The sensors will be mounted on the fixed ends, in a way that no distortion or error should occur in the measurement of distance due to the signals sent by any other sonar sensor and at the same time target boards will be used at the opposite ends of each distance sensor for accurate reading.

Sonar sensors emit sound waves and there is a likely chance that misreading’s could occur. However, if we carefully plan on where to place the sonar sensors and perfectly direct the target opposite the distance sensor, then that chance can be minimized significantly.

The Monitor module communicates with the middleware through UDP, it has a Python script running on it which commands the chair to move in specified positions.

A circuit board

Description generated with very high confidence

Fig 4: Sensor Server Connection

Fig 4 shows how the sensors are connected to the mega board using the adaptor board and the USB cable which is used to connect the boards to the PC to upload Arduino sketches.

Testing

To do a testing of the system created the following steps had to be done

Step 1:

First, we had to do a testing for the sensors where we had to test whether the sensors were working according to the data sheets and giving the correct output. to do that we connected the six distance sensors to the adapter board and ran the Arduino sketch to observe the output data. If the sensors were not connected properly a zero will be displayed for that sensor to specify that no data is coming from the sensor.

Step 2:

The next thing was to test whether we were receiving the correct sensor data which was done by running an Arduino sketch on the Uno which displays both the distance and angular data on the console to observe the sensor data read.

Step 3:

To test whether the Monitor was receiving data from the middleware and from the Sensor Server, a python script created for the monitor was ran and the output results were observed where the difference between the two data values were to be displayed in a file with a .csv extension.

Step 4:

To the test the final functionality of the whole system, this was done by doing all the connections needed and running the system scripts and sketches as described earlier and observing the output results.

For further details on testing see the test plan document.

A screenshot of text

Description generated with high confidenceThe following figure shows an example of the sensor server output when the testing took place.

Fig 5: Sensor Server output

Results

Lessons Learned

There is always a lesson to learn when one is doing a project, the lesson can be learned throughout the project period or after one is finished with the project, in our case the following are the lessons that we learned during this time when we were doing the project.

There were many requirements that had to be made in order to complete this project. Although most of the requirements were met, there were a few things that we thought needed to be improved such as time management, organizational skills, communication skills and technical skills.

Time management – at the beginning of this project we had a plan of what the project required us to do but the concept of it at the beginning was not clear enough and so it took us quite a few weeks to get a hold of what needed to be done and in doing so the time went quick and we were not able to manage our time properly thus wasting a lot of time.

Certain tasks were not completed on time because they were interdependent and some of the group members were not able to complete the tasks on time.

Another thing that made us waste a lot of time was that we did not anticipate that one of the group members was not going to fully commit to the project till the end and in doing so it came to a point where he was not able to continue working on the project and thus some tasks were left incomplete and we had to use more time to complete those tasks. Thus, we should always plan a head and anticipate for such occurrences.

Organizational skills - when doing this project, we came to realize that organization was a key to a successful project. In the beginning of the project we did not have proper organization of our work and in doing so some documents got lost and thus cost us a lot of time which we could have used for other important tasks. As things got more complex and we had a lot of documentations and coding to do a better way of organizing ourselves had to be implemented and thus GitHub came in to use. We also used a bit of google docs at the beginning but we opted to use GitHub which made it easier for us to share and work simultaneously on the documents and coding.

Communication skills – at the beginning of the project we had a lot of issues with communication among the group members. Some of our team members were unable to participate in meetings which were held outside the class time. Also, some team members were not responding to emails and text messages which needed them to complete certain tasks, and in doing so it made us fall behind. For better communication we had to encourage each other to respond to emails, attend meetings and take responsibility for the tasks they had been assigned to do.

Technical skills – this project required a lot of technical skills which some of our team members lacked initially. To improve these skills, we had to learn as we progressed with the project and in doing so we were able to develop new skills and enhance the skills we already had.

With all these challenges we came across we have managed to complete the project and learn from those challenges. Unfortunately, it is not always possible to complete all the tasks that you have set out at the beginning on time, as there are factors you don’t take into consideration at the beginning that pops up later.

Conclusion

This project took us about 24 learning weeks to complete and within those weeks we were able to design and build a system that measures the accuracy of the 6DOF platform by comparing commanded data with actual sensor data and displaying the results. Throughout this project we were able to learn a lot and develop our skills in terms of technicality and theoretically. This report covers the process that was taken for a successful project and how it was implemented. The report also involves an overview of some of the documents that will be included within this document.