University of South-Eastern Norway

Institute of Science & Industrial Systems,
Kongsberg

NAGINI

Systems engineering management plan

by

Mehtab Singh Virk
Sameed Ahmed
Han Zhou

June 1, 2020

Abstract

This document outlines a systems engineering management plan (SEMP) for the development of a robotic snake; Nagini. An accompanying project management plan (PMP) and prototype documentation, can respectively be found in [1] and [2].

Table of contents

List of Figures

List of Tables

1 Technical Summary

This document provides insight and overview to the engineering of system-of-interest; Nagini. It was given as compulsory semester-project in ”ES-SES4000: System engineering” 2020 to Mehtab Singh Virk, Sameed Ahmed and Han Zhou by supervisors Steven Bos and Henning Gundersen, University of South-Eastern Norway (Kongsberg). System-of-interest is a robotic snake. Purpose of system-of-interest is to follow an object at a given range. Design of system-of-interest is entirely by project team. An accompanying project management plan (PMP) can be found in [1], and a prototype documentation can be found in [2]

1.1 Summary of system-of-interest

In order to move along its robotic body, a DC motor actuates the frontmost wheel beneath system-of-interest's head. Horizontal turns are made by servo motors mounted onto head of robot and each link; i.e. each turnable point of the snake. Each link is connected to the next through a universal joint that grants freedom of movement in approximately 360 degrees. An Arduino Mega2560 fulfills the role as the robot's brain; behaviour of system-of-interest is controlled through programming of stated microcontroller. Non-electronic components are either easily obtainable Lego modules or 3D printed parts. A complete list of utilized parts can be found in Appendix B.

As of June 1, 2020, the developed prototype is partially functional with existing software. In order to complete the prototype for all its intents and purposes defined by stakeholder and system requirements (see section 1 in [2]), slight progress in software is needed.

2 Introduction

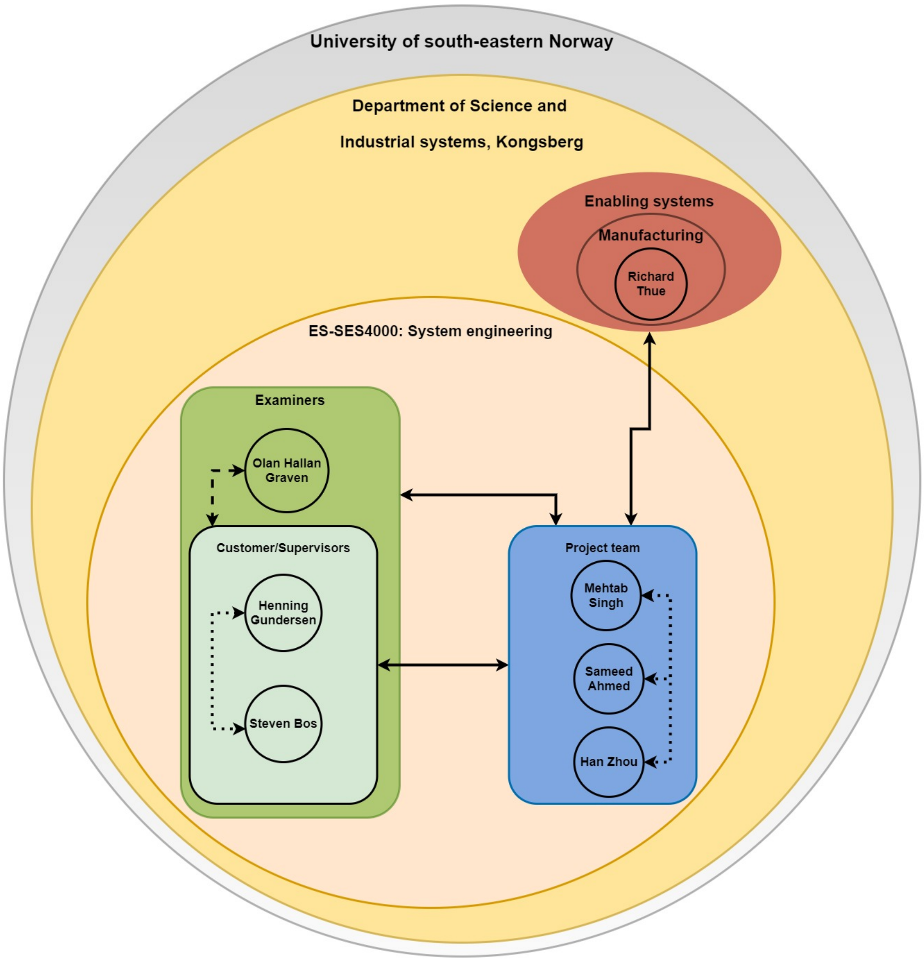


Figure 1: Relational structure of essential entities in project-of-interest

See Fig. 1 for relations building the fundament of project-of-interest's structure. Project-of-interest is a part of compulsory coursework in "ES-SES4000: System eng." present in syllabus of course in "Systems engineering with Embedded systems". The course is conducted by "University of south-eastern Norway", specifically by the "Department of Science and Industrial systems, Kongsberg". Project supervisors are Henning Gundersen and Steven Bos, from whom project-of-interest was acquired.

Supervisors and institute leader Olaf Hallan Graven constitute the examiners that will set a grade on this project, its execution and completion. In addition to that, Richard Thue has contributed with manufacturing of project team's designed parts used in Nagini and thereby classifies as an "enabling system" in accordance with [3].

Moreover, the mission of project-of-interest is to produce a prototype capable of moving from designated point A to B, which contextual base will be presented in section 2.1. Project team has named system-of-interest "Nagini", meaning *serpent* in Indian languages. See [4] for further information regarding history of the term.

2.1 Basis Context

The unprecedented and accelerated rate at which technology is developed since 1970s, has given critical opportunities for greater planning and execution of safety-critical missions that otherwise would have been complicated and complex, albeit not impossible, to carry out without existing tools. Accordingly, as many scientific areas and newer expertise has begun utilizing artificial intelligence in modern systems for problem solving in various fields, search and rescue (SAR) missions have since 2000s contributed to the overall requirement database and concept analyses that is needed to further develop systems that can aid in SAR operations. Many such systems are robotic.

In this age, autonomous airborne robotic drones with swarm- or hivelike features can soon be standardized for SAR missions (see [5],[6],[7],[8]). However, robots made in image of certain animals and reptiles have greater chances to succeed in SAR missions due to their advanced and adaptive gait that lets them traverse otherwise uneven, complex and perhaps dangerous terrain during emergency states. Alligators, crocodiles, lizards, and turtles are the most common reptiles, in addition to *snakes*; the main category of limb- and legless vertebrates.

Customer and supervisors Henning Gundersen and Steven Bos has approached project team with the mission of developing a prototype snakebot(i.e. system/product-of-interest, see [9]) that can move from a arbitrary point A to point B on a horizontal plane. Project team has chosen to implement the high-level requirement in terms of a object-following robot. See Fig. 2 for reference.

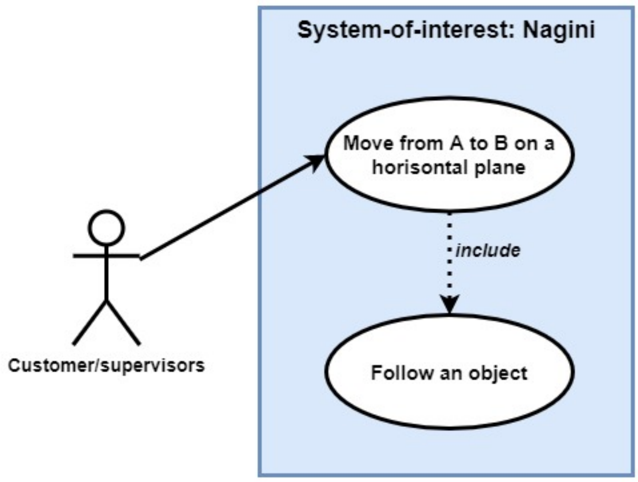


Figure 2: Use-case diagram showing high-level functionality of Nagini

2.2 Operational Context

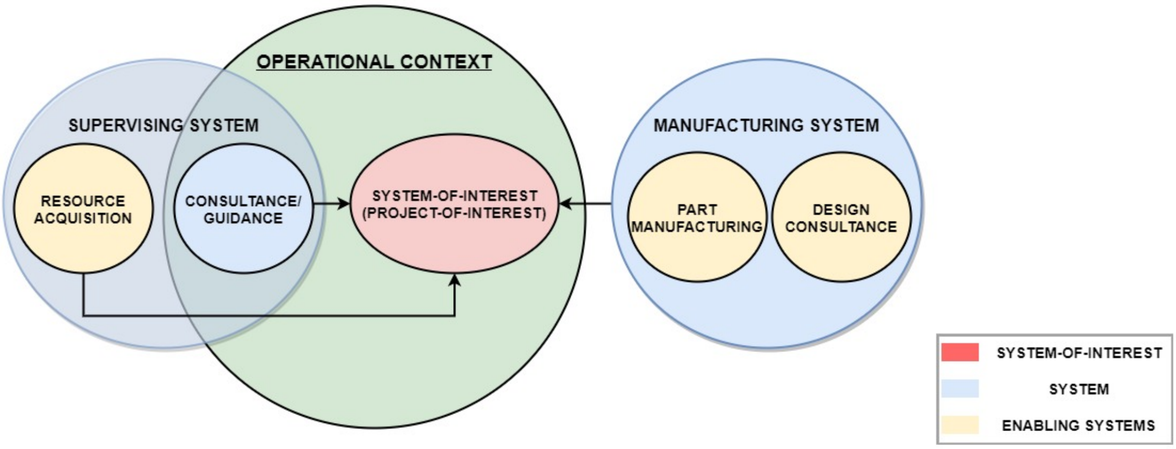


Figure 3: Operational context diagram showing system-of-interest and supporting systems

Fig. 1 depicts the operational context of project-of-interest, i.e. the context in which supporting systems to some degree contribute to the operation of a system-of-interest. Our system-of-interest is in this context project-of-interest. Per [3], all subsystems of manufacturing system and resource acquisition subsystem of supervising subsystem classify as enabling systems; services that enable progression during a stage of system-of-interest's life cycle. Resource acquisition subsystem is essential for gathering needed materials during design phase, but otherwise invaluable in further life cycle stages. Likewise, the manufacturing system might be needed for discussing, improving designs and manufacturing feasible parts, but at prior or later stages this system is irrelevant from the perspective of project-of-interest.

While enabling systems are not considered to be a part of the operational context of system-of-interest, otherwise supporting systems are. In other words, the project-of-interest would not have been able to operate/execute at all if not for the consultance subsystem of supervising system. Note the overlap between supervising system and operational context for project-of-interest in Fig. 3. The contributing factor for project-of-interest's operational context is therefore the consultance subsystem. If not for the contributing factor, the system would have been classified purely as enabling system.

3 System-of-interest Description

The developed system-of-interest is designed according to stakeholder and system requirements defined in section 1 in [2]. The purpose of system-of-interest is to follow an object at a specified range. It does that by actuating (DC motor) a frontmost wheel when an object is detected (HC-SR04 sensor). In addition to that, the system-of-interest tries to turn (servo motors) if an object is detected at the specified range, but at an angle. For initial construction plan including models of components and other designs, see Appx. A.

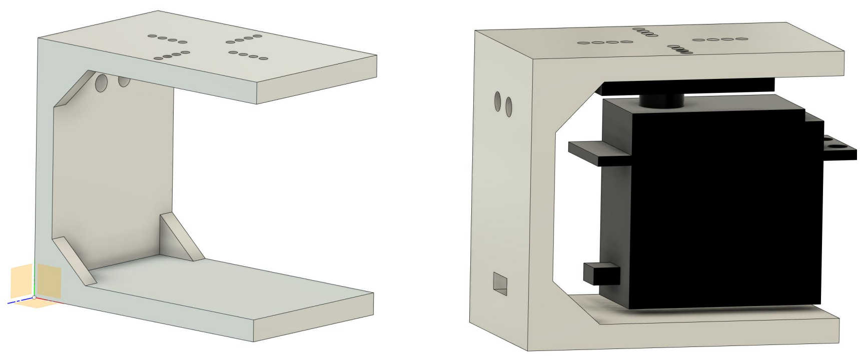
This description of system-of-interest aims to describe necessary background theory, the used hardware, the software, and the coordination of the two. See Appendix. C in [2] for a connectivity diagram showing connections between electronic components. See Appx. D in [2] for the master Arduino code implemented in latest build of system-of-interest. Alternatively, see Github repository in [10].

3.1 Background Theory & Design Implications

Per [11], the kinematic parameters of a snake robot can be modeled as shown in Fig. 2.1 by parameters defined in Table 2.1, p. 41. From a hardware design perspective, this gives the opportunity to design several solutions that will fit the given kinematic description. In essence, a functional design needs links, a means of turning them and propelling forward. Based on this, project team decided to go with servo motors to turn said links and a DC motor to actuate a single, leading wheel. The links and joints presented are functional and designed by project team.

3.2 Hardware Description

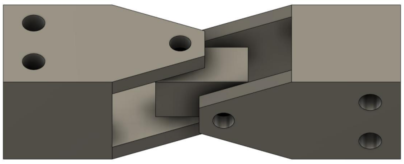
The hardware for system-of-interest consists of custom-designed 3D printed parts, some standard lego modules, screws, bolts and easily obtainable electronics. See Appendix B for the total equipment used in the latest build of system-of-interest. The listed 3D printed parts are downloadable. To download them, click their respective names in Appendix B. Print these designs using a functional 3D printer to manufacture parts.



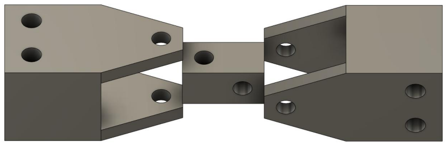
4.1: Link v0.1 without servo motor 4.2: Link v0.2 with servo motor mounted

Figure 4: Link designed for rotational capabilities

There is a majority of electronic components capable of producing rotational motion. The project team utilized TowerPro MG995 servo motors as means of rotational movement since those were suppliable by resource acquisition system. See section 15 in [1] for further information regarding resource acquisition system. A design for a link that could implement this electronic component was created. However, a design that would house the servo motor without bottom space was not efficient since the link needs to turn by the rotational means of a servo motor, without frictional force from encapsulated motor. This gave rise to the links depicted in Fig. 4.1 and 4.2. Note the space between link floor and motor in Fig. 4.2.



5.1: Universal joint ready to be assembled.



5.2: Universal joint constitutes two arms and a central brick

Figure 5: Universal joint v0.4

Furthermore, to implement freedom of movement between links, a universal joint has been designed. The universal link grants approx. 360 degrees of motional freedom and is necessary in development of vertical motion for future versions. For future development plans, see section 5. Moreover, to let the robotic snake move freely along its length, passive set of wheels made of lego were attached underneath each link. See Fig. 6 as reference. Fig. 6 also shows the first development iteration of system-of-interest where lego joints were used instead of universal joint v0.4.

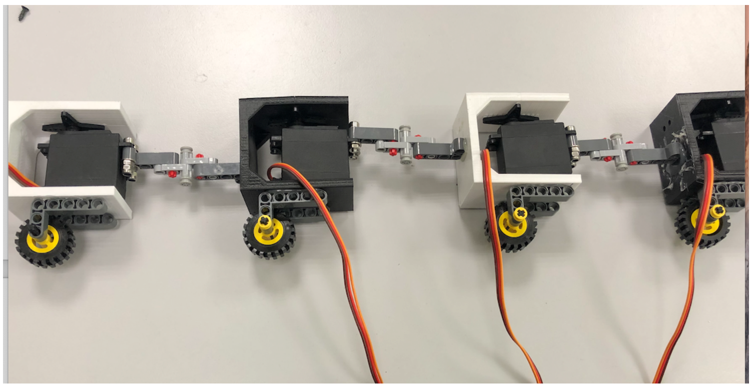


Figure 6: Links with joints v0.1 made of lego

In addition to that, two versions of the head which would encapsulate electronic components such as motor driver, battery, Arduino and servo motor were designed. See Fig. 18 and 19 in [2] for illustrations of head v0.1 and Fig. 20 and 21 for illustrations of head v0.2. Unfortunately, these designs were not sufficient as they were unable to rotate due to servo motors incapability to rotate such large objects. This resulted in a last-minute head v0.3 which is produced by following step 7-8 in section 8.2 in [2]. See Fig. 7 for a illustration of head v0.3. Since head v0.3 is not as big as v0.1 and v0.2 in terms of dimensions, it is able to turn. However, this resulted in the sudden change of component placement. Since head v0.3 is not big enough to encapsulate all components, remaining components have been attached to top of connected links with velcro or double sided tape.

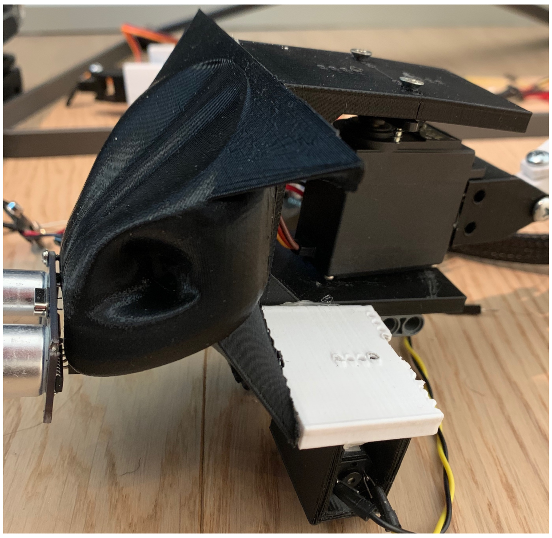


Figure 7: Head v0.3

Fig. 7 also shows distance measuring sensor HC-SR04 mounted onto head v0.3 and the DC motor in compartment. This introduces the capability to determine if an object is at a specified range in front and propelling system-of-interest forwards or backwards.

3.3 Software Description

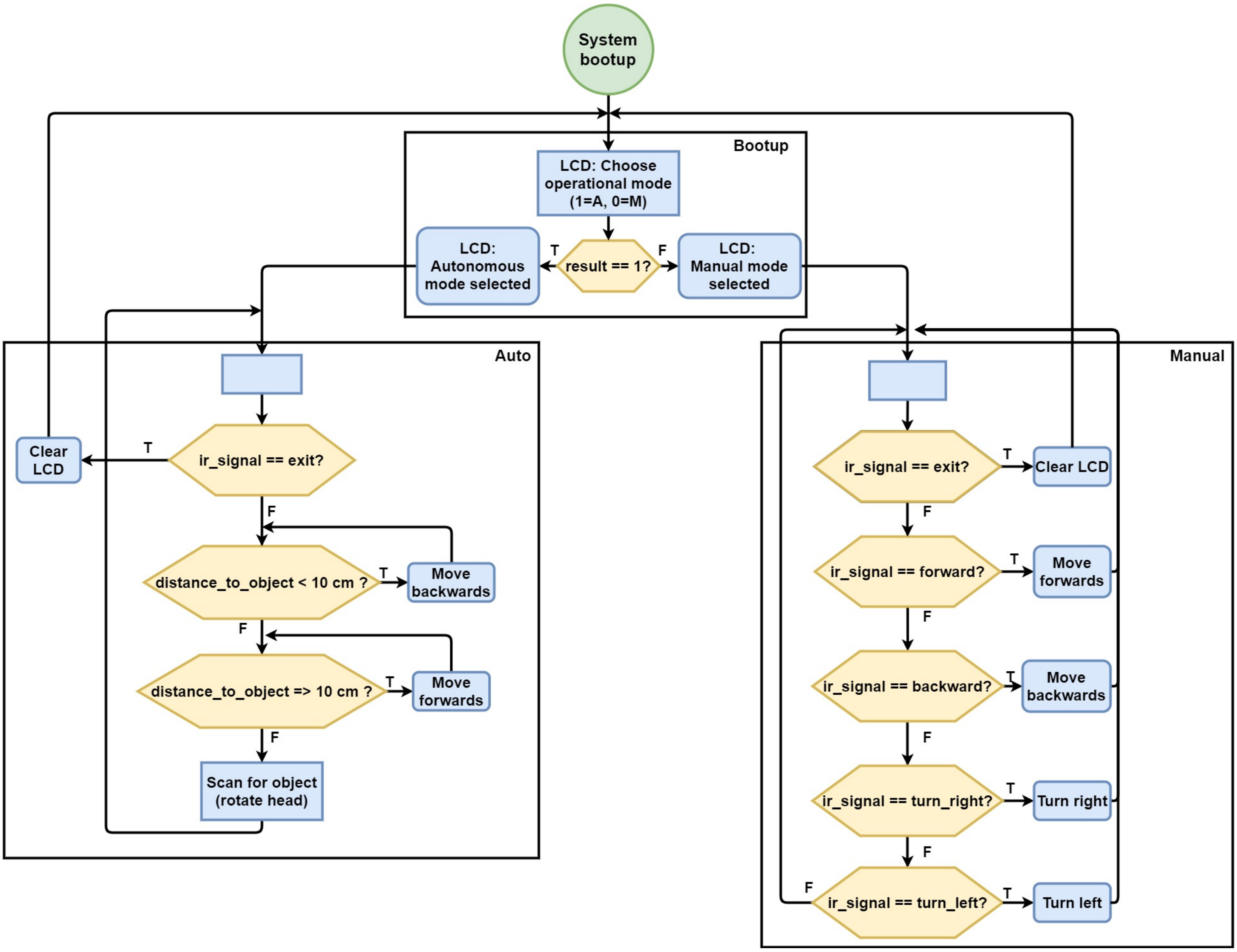


Figure 8: ASMD chart showing decisional workflow of control path

See Fig. 8 for a description of system-of-interest's (i.e. Nagini's) decisional workflow. After system bootup, control path (i.e. Arduino subsystem) displays a message for the user on LCD subsystem; "Choose operational mode (1=A, 0=M). Afterwards, control path checks whether a input signal is equal to a logical '1' or '0' as depicted in "Bootup" state. See [12] and [13] for more information on Finite State Machines, and [14] regarding Finite State Machines with Datapath. If the infrared input signal is a logical '1', the control path transitions onto "Auto" state.

In "Auto" state, control path determines if there is an input signal corresponding to "exit"; if so, the control path clears the LCD subsystem and transitions back to the "Bootup" state. Otherwise, the control path continues to check whether there is lower than 10cm distance to an object; if true, system-of-interest moves backwards in order to increase the distance to object and checks the Mealy statement again. If the Mealy statement returns false, the control path continues to next Mealy statement in order to check whether the distance is greater than or equal to 10cm. In case that is true, the system-of-interest moves towards the object to decrease the distance and checks the Mealy statement again. Otherwise the control path executes Moore statement, i.e. it turns the head in order to survey its surroundings for objects. Thereafter, it loops back to same state again to repeat flow.

On the other hand, if Mealy statement in "Bootup" returns false, the control path transitions to "Manual" state. The state box is empty so control path continues to first Mealy statement; checking whether a infrared signal for state-exit has been received. If not, control path continues towards second Mealy statement. In this statement, the control path determines whether a signal for moving forward has been received. If so, the system-of-interest moves forwards. Otherwise, the control path continues to check whether different input signals have been received, and in case they have, the corresponding action is executed. As seen in Fig. 8, this continues until the control path reaches the last Mealy statement in "Manual" state. In case a signal for turning left is received, the system-of-interest will turn left, otherwise the control path loops back to the beginning of same state in order to loop behaviour.

4 Project Planning

See Fig. 9 for the general life cycle model of a project.

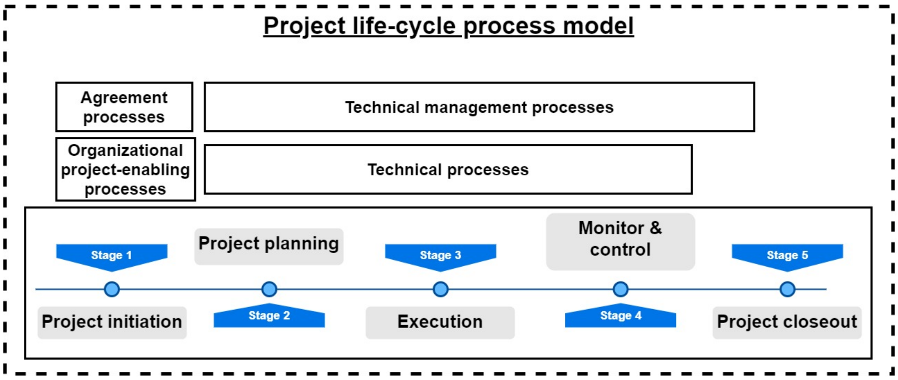


Figure 9: General project life-cycle with project management processes

Usually for any task, planning is done initially. Then task execution pursues and wanted result is purposefully gained. However, planning in projects is a flexible, continuous practice that results in a more refined plan for each planning iteration; that is, project planning is not to be interpreted as an initial task, but a continuous, iterative practice that further develops for each time. Although, an initial plan for project-of-interest is essential to begin. An iterative approach to planning leads to higher chances of project success. See section 2 in [1] for further information on the project planning scheme utilized by project team throughout project-of-interest's life cycle.

4.1 Project deliverables

The project-deliverables are a systems engineering management plan, project management plan and prototype documentation. See table 1 for further reference.

Table 1: Project deliverable with delivery- date and medium.

Project-deliverable	Delivery date	Delivery medium
Project Management Plan	12:00 AM 02/06/2020	Online (through Wiseflow)
System Engineering Management Plan	12:00 AM 02/06/2020	-
System (i.e prototype) Documentation	12:00 AM 02/06/2020	-

4.2 Project work plans

4.2.1 Work breakdown structure

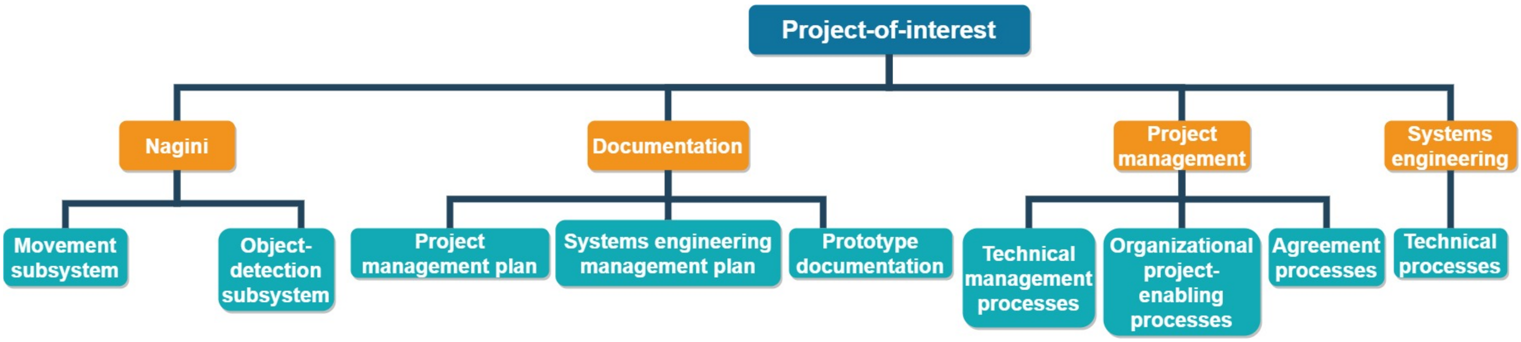


Figure 10: Work breakdown structure (WBS)

See Fig. 10 for work breakdown structure of project-of-interest. For further information on work breakdown structures, see [15] and [16]. On left-hand side in Fig. 10 the project-deliverables "Nagini" and "Documentation" are depicted, while system-of-interest and project-of-interest process groups are to the right. For more information on depicted process groups, see p.24 in [17]. Please note that 2.level (blue) entities has to be integrated together to constitute their respective 1.level deliverables.

The work breakdown structure tells what artifacts will be needed for the project-of-interest to be successful. A way of interpreting the work breakdown structure is "in order to deliver project-deliverables on left-hand side, strategies and activities on right-hand side have to be practiced".

4.2.2 Network plan: First prototype build

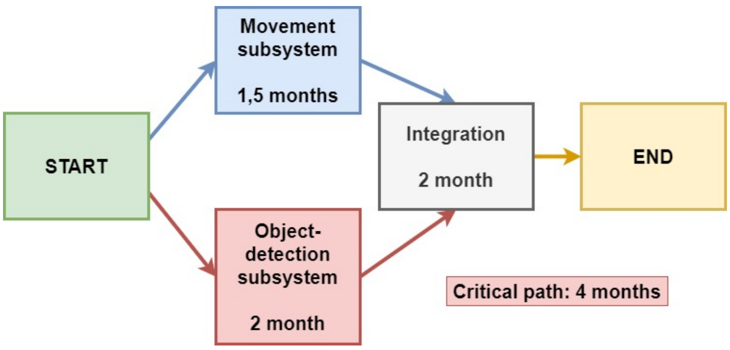


Figure 11: Network diagram for first shippable prototype

See Fig. 11 for a network diagram depicting the critical path (longest path from start to finish). For more information on critical path, see [18], [19] and [20]. This diagram is for the development of the first feasible prototype. The critical path is 4 months, meaning that is how long it will take for the first prototype to be designed. Any delays to the critical path will delay the whole prototyping. The network plan takes into account complications imposed by SARS-CoV-2, described in section 15 of [1]. However, bear in mind that this is an initial estimate that is not absolute. Wanted or unwanted emergences affect planning and progress of project-of-interest positively or negatively. It is therefore imperative that team communication is good, and due dates are met.

5 Future Development

In order to extend functionality of object-tracking, a Pixy2 module can be used with existing Arduino to implement better algorithms for object-following purposes. See [21] for more information. This introduces machine learning algorithms that are more robust for system-of-interest's purpose, by e.g. adding the ability to only track a predefined object and avoid others.

Adding capability of an onboard battery pack allows system-of-interest to be independent of cables attached to external power supplies such as PC and power banks. However, an added battery pack has to supply enough power in order to power all connected components. To properly add this capability, a weight analysis with the potentially added battery pack should be conducted. This will distinguish how much space and where the battery pack is suited for mounting.

A more robust and reliable means of wireless communication than infrared should be pursued in order to increase overall system reliability. Examples of such communication frameworks are bluetooth, wifi and radio frequency (RF).
In addition, a new face can be developed that encapsulates the distance measuring sensor. This will give protection to the component while reducing overall external space utilized. A new head that is lightweight enough for turning, but can house the DC motor is also a possibility. As of June 1, 2020, head v0.3 has to be manually created by modifying existing parts. This can be avoided by a new design. Furthermore, the links can be updated so that the lego technic beams and link is 3D printed attached. This will lead to fewer lego parts. Larger wheels attached to head and each link will contribute to a better ground-to-body clearance, which is a problem in the latest build.

Last but not least, adding the functionality of vertical motion will introduce three dimensional movement of system-of-interest which can further extend its potential usage in later versions to SAR missions described in section 2.1.

6 Budget & Resource Management

The Project Leader is going to be responsible for managing and reporting on the project's cost throughout the duration of the project. The Project Leader will present and review the project's cost performance during the monthly project status meeting. The project leader is responsible to calculate the cost deviation and put forward the option to the project Supervisors how to put project back on budget, earned value calculations can be used for this purpose. Snakes Robot project sponsors are responsible for all the budget authority and decisions and to incorporate budges changes.

Cost and Schedule Performance Index are going to be reported on a weekly basis by the Project leader to the Project Ssupervisors. Variances of 5% or greater within the cost and schedule performance will change the status of the value from Green to Yellow which is cautionary. These are going to be reported and if it's determined that there's no or minimal impact on the project's cost or schedule baseline then there could also be no action required. Cost variances of 25% or greater within the cost and schedule performance indexes will change the status of the value to red or critical. These are going to be reported and need corrective action from the Project leader so as to bring the value and schedule performance indexes back in line with the allowable variance. Any corrective actions would require a project change request and be must be approved by the both Supervisors before it is often implemented.

7 Engineering Disciplines

Engineering disciplines are mostly component-oriented and value-neutral in their intellectual content, see [22]. Whereas system engineering is boarder than component oriented. How the different domains and engineering disciplines are inter-correlated in this project are illustrated below in Fig. 12. See [23] for further information. This diagram shows how the whole Snake robot project has been implemented by involving different disciplines.

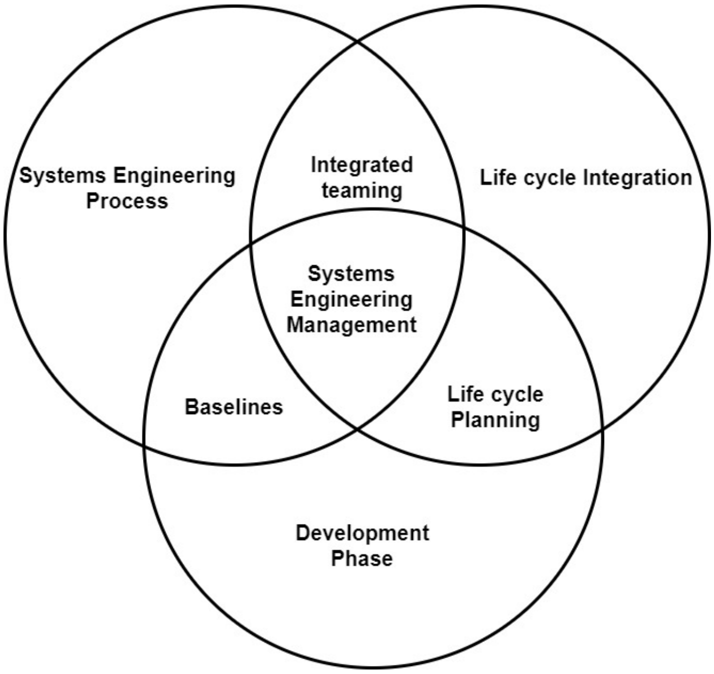


Figure 12: Relationship and overlap between different disciplines

8 Logistics Support

The technical process and management through which logistic support and supportability considerations are integrated into the design of equipment or system is known Integrated Logistics Support (ILS). It is the process by which all elements of logistic support are planned, acquired, tested, and provided in a timely and cost-effective manner, see [24]. The illustration for integrated logistics support is given in Fig. 13, also see [25].

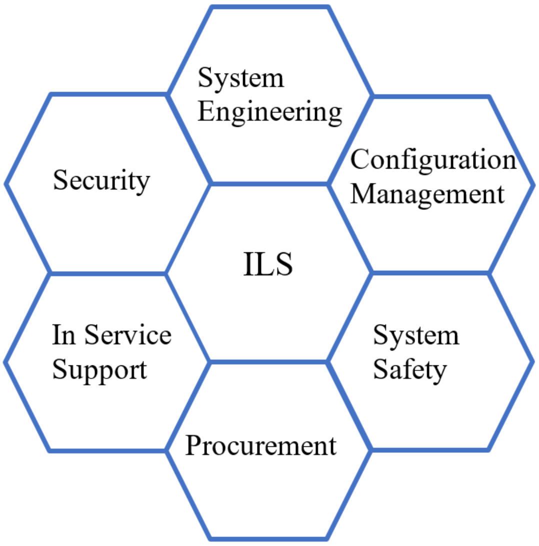


Figure 13: Integrated logistics support.

9 Quality Assurance

All members of the Snake robot project team will play a part in quality management.

The Project Sponsor (Both Supervisors) will review all project tasks and deliverables to make sure compliance with established and approved quality standards. The responsibility of project sponsor also includes the acceptance or rejection of the project deliverables. While the Quality Management will be responsibility of the project leader from the start until the completion of the project. It will be the responsibility of the project leader to implement the Management plan for quality assurance and make sure that all processes, documentation and tasks are aligned with the plan. The Project leader will work with the project's quality specialists to determine acceptable quality standards. The Project leader is additionally liable for communicating and tracking all quality standards to the project team and stakeholders.

Quality Specialists are liable for working with the Project leader to develop and implement the standard Management Plan. Quality Specialists will recommend tools and methodologies for tracking quality and standards to determine acceptable quality levels. Quality Specialists will create and maintain internal control and Assurance Logs throughout the project.

The Quality assurance specialist and project manager are assisted by the stakeholders within the establishment of acceptable quality assurance standards, while the remaining team is going work to make sure that each of the quality standards are met and communicate any concerns regarding quality to the Project leader.

Quality control for the Snake robot Project will utilize methodologies and tools to make sure that each deliverable of project results in approved quality assurance standards sets at the start. To satisfy deliverable requirements and expectations, we must implement a proper process during which quality standards are measured and accepted. The Project leader will ensure all quality standards and internal control activities are met throughout the project. The project leader by the assistance of the Quality Specialists (other project group members) will verify that each quality standard set by the organization is met for every deliverable.

If any changes are approved after the proposal by the Project Sponsor, the Project leader is liable to convey that changes to the project team and updating all documentation and also project plans.

10 Configuration Management

Project Configuration for the Snake Robot Project goes to be created using MS Project 2019 starting with the deliverables identified within the project's Work Breakdown Structure (WBS). Activity definition will identify the precise work packages which must be performed to end each deliverable. Activity sequencing goes to work out the order of work hours packages and assign relationships between project activities. Activity duration estimation can be used to calculate the amount of working hours periods required to end work packages. Resource estimation can be used to assign resources to work packages to finish schedule development.

The project team will tentatively assign any resource to the project task when the preliminary configuration management plan has been reviewed after the development phase. The project team and resources must meet the proposed work package assignments, durations, and configuration management plan. Once this is achieved, the project sponsor (Supervisors) will review and approve the configuration management plan and it will then be baselined.

11 Knowledge Management

Knowledge is often divided into tacit and explicit, see [26]. In Knowledge management, three things are closely connected information, data and knowledge.

In this Project, the project leader is going to be liable for Knowledge Management with the project team. The project leader will arrange Social events, formal and informal discussion meeting to market knowledge management among the team.

The proposed schedule is reviewed and approved by the project sponsor before it baselined and the assistance in validation of the proposed schedule and reviews is the duty of the project stakeholders.

12 Project Development Plan

The Main reason for project development planning is to force people think before acting, allow feedback between actual state and goals and schedule usage of scarce resources. The Pert chart for the Snake Robot Project is given in Fig. 14.

Activity	Description	Predecessors
A	Start Development of SEMP	
B	Project introduction and Scope	
C	Budget and Resource Management	B
D	Project Planning	A, C
E	Risk Management	D
F	Knowledge Management	C
G	Logistics support	F
H	Quality Assurance	F
I	Engineering Disciplines	B
J	Configuration Management	H, E, G

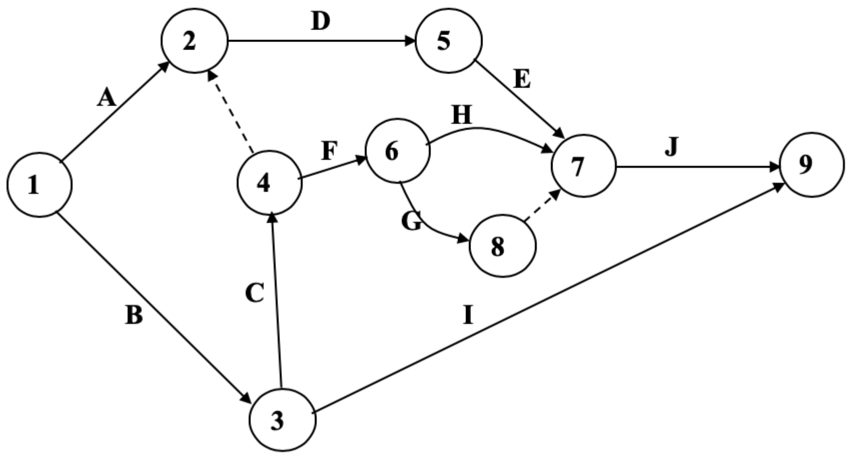


Figure 14: PERT chart

12.1 Considerations

Following is the few important steps which we considered during the designing and development of a system that has store important information or data.

Encrypt has been used for any sensitive information that would cause serious damage to the project if it were lost.

If any personal information or data is lost accidentally can be recovered in order to prevent any damage to the project.

Continuously updation of our security measurement has been done which can helped us to keep the sensitive data more secure.

Another Important point was our computer security which is connection to our organization need to be appropriate in order to prevent serious damage to our organization as well as to the project.

Another good approach used regularly in our project take the back-up of the information stored on the organization computers (USN)and also keep them in different places in this way if you loss your computer or your data lost accidentally, you don't lose the information.

12.2 Reflection on Gantt Chart

The evidences which are provided as a part of this chapter above, are the initial pieces of the work. We have drafted it at the start of our Group project. We used it because we feel that we have more or less followed the same time schedule with some minor modifications. These modification occurred due to the mistakes which we have made. These mistakes are discussed below.

During the course of our group project, We have faced some difficulties related to following the schedule of Gantt chart. As the management of time is everything in the Project but ups and downs are also part of every project. The biggest challenge was that when an activity was finished before time, it was very hard to start the other activity early because of the other task needed to be finished first before the next task can start. Similarly when an activity took a more completion time than its actual completion time, that can also cause delays in the project time But this kind of things only happened at most once or twice during the project.

Secondly, We have faced some challenges at managing multiple tasks at the same time. But, during the past months or so, we have found it very difficult to perform multiple jobs at the same time. Evidence shows in our weekly routine of managing our project. It can be clearly seen from the above provided evidence that we have dedicated a multiple day for carrying out the different activities dedicated for the management of our project. Though Gantt chart provided us with the insight about which task to do in a particular week but, it lack the sequence of activities that needed to perform on that particular day.

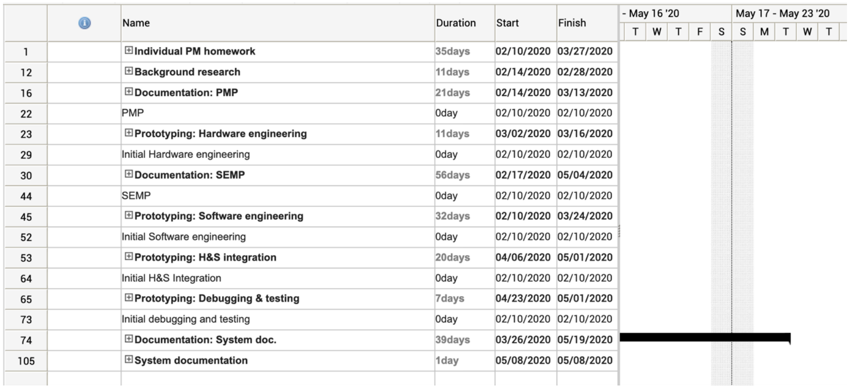
Specific:
1. Update Gantt chart on a regular basis. Start with 5 day intervals for larger projects. Assess the time intervals after a few projects. 2. If doing multiple project tasks at one time; create a detailed timetable for those tasks.

Measureable:
It can measurable by how efficient we were during the span of our project by checking If work has been done more efficiently then the system is working.

Achievable:
We have the tools to be able to timetable and update activities regularly.

Realistic:
Although we have dedicated sometime in creating and updating our project timetables, it will ensure how much efficient we are. The goals set out here are realistic as long as we have the determination to updating.

Timely
The system should be employed on all future projects. Gains should be made on each project.



	Name	Duration	Start	Finish
1	**Individual PM homework**	35days	02/10/2020	03/27/2020
12	**Background research**	11days	02/14/2020	02/28/2020
16	**Documentation: PMP**	21days	02/14/2020	03/13/2020
22	PMP	0day	02/10/2020	02/10/2020
23	**Prototyping: Hardware engineering**	11days	03/02/2020	03/16/2020
29	Initial Hardware engineering	0day	02/10/2020	02/10/2020
30	**Documentation: SEMP**	56days	02/17/2020	05/04/2020
44	SEMP	0day	02/10/2020	02/10/2020
45	**Prototyping: Software engineering**	32days	02/10/2020	03/24/2020
52	Initial Software engineering	0day	02/10/2020	02/10/2020
53	**Prototyping: H&S integration**	20days	04/06/2020	05/01/2020
64	Initial H&S Integration	0day	02/10/2020	02/10/2020
65	**Prototyping: Debugging & testing**	7days	04/23/2020	05/01/2020
73	Initial debugging and testing	0day	02/10/2020	02/10/2020
74	**Documentation: System doc.**	39days	03/26/2020	05/19/2020
105	**System documentation**	1day	05/08/2020	05/08/2020

Figure 15: Gantt Chart

13 Improvement Plan

The Project leader, has the overall authority and charge for managing and implementing and improving this project according to this Project Plan and its Subsidiary improvement Plans. The project team will consist of personnel from the quality control/quality assurance group, coding group, hardware engineering, software engineering, technical writing group, improvement group and testing group. Project improvement planning will be performed by Project leader with all resources available for the project. All project and subordinate improvement plans are going to be reviewed and approved by the project sponsor (Both Supervisors in this case). All funding decisions also will be made by the project sponsor. The Project Leader and Project Sponsor both are also responsible for the signature of approval authority for the writing documentation.

The project team are going to work in a matrix. In teams, different members from each organization still report back to their organizational management throughout the duration of the project. The project leader is liable for communicating with organizational managers on the progress and performance of every project resource.

When we first embarked on our group project at the start of year 2020, We had limited information and knowledge about software design patterns. Over the course of 4 months, we have learnt many different software design patterns that had help us to create better applications for the snake robot.

13.1 Reflection

Specific:
To continue to develop a greater understanding of software design patterns.

Measurable:
To measure how well we have interpreted requirements, answer the following
1 - How easy is it to maintain and update the system.
2 - Can the system be applied to companies with a similar essence.
These questions would also measure the improvement of our design pattern skills.

Achievable:
When compared to previous developments, the new developments will consist of more design pattern and there are more components that can be reused.

Realistic:
To understand all the behavioral design patterns within 4 month period.

Timely:
To make small improvements in each development.

14 Information Management Plan (ISO15289 [27])

14.1 Github

It was a suggestion to use Github to manage the code information of a team and to have the overview of the code development progress. Github is a popular open source platform to manage and share the information flow of the software development. When there is the information sharing, there will be security issue. If we wanted to have control of our project content, the software project leader shall have the ownership of our Gitbub resources.

In Github, there is setting of repository for private and public. When the repository is set to private, then the code information will not be shared with others, even those follow your repository. When the repository is set to public, then each of the team member that is invited by the project leader, can access it. Meanwhile the Github itself has a system to register the development process for you, each time you do a commit, either in a master or in a branch.

There will be an updating which the project leader can have the overview. This will fulfil the ISO15288:2015 [17] process of information management, according to ISO15289 [27].

14.2 Acquisition of resources

When we acquire the resources of the software information, a process will be started with setting up a resource list, such as the drawing size and form, the connectivity of the system, and the component list of the system, so that the software and the hardware can be worked together. This will make it easier for us to reach a successful acquisition and to manage the resources in the documents.

In an acquisition plan, we need to think over the tasks need to be done, the difficulties and complexities. How difficult is it? If the acquisition is from some suppliers, it will be a lot of communication with the supplier to discuss and to negotiate. Of course, we can also buy online from ebay or Amazon. It will be an advantage to take time and making a good plan, so we know how to do it and what is needed. We shall also use some Microsoft tools to make the acquisition plan and to follow the acquisition process. This is also a part of the information management according to ISO15288 [17] acquisition process.

For example, if we acquire code from Github, it is vital to register the code deployed in our system is the original code or the modified code. And shall citate the code and software solutions that is from Github.

15 Information Security Management Plan (ISO15289 [27])

During the project, we met coronavirus pandemic of that reason, most of the collaboration were online. Meanwhile, as of today, cyber security is a big problem for many worldwide. It is not just every one who work by online is aware of such security problem, like spam, virus, hacking, false email, false webside. USN has strong recommendation to the student to be aware of the false webside. It is important to check all the time that the webside you browing is the real one, the correct site. Hackers today not just misleading the users, use false webside to trap the users, they also use the false website even the official website to steal the personal information from the user, and the communication content from the communicating partners. The edge technology, the wire usage and the mirror techniques still have big potential to failed by the unwanted information. Big companies like Microsoft and Google are updating their software all the time. Thus we follow the ISO15289 [27] information security plan, use ZOOM meeting place as our online information resources from this aspect.

15.1 Zoom

We used ZOOM to communicate, to have group meeting. In ZOOM system, the security process make it possible that only the authentic one has the availability of the meeting. Participants shall get invitation to get access to the meeting room. Furthermore, an unregistered or unsecured meeting invitation shall be sent at least 24 hours before the meeting start. Otherwise, the availability to the ZOOM meeting arrangement will be denied and failed. And the host control the meeting, as in the real world, even more control which normally is not so easy to manage only by meeting culture. That is a participant who has no identity registered will be presented as guest, so it is easy to identify who is registered, who is not. The host can secure the meeting information, both in video and in text.

However, there is inconvenient weakness of the ZOOM. For example, we cant send an ZOOM invitation directly with an agenda. The participants have potential possibility to disturb the information sharing during the meeting, that some important information may unintentionally be removed, lost or changed. To avoid this, the users of ZOOM meeting shall be aware of this weakness and respect the meeting procedure the host required.

With regards to information security, we shall have the knowledge of the key length verses strong security. We have learned from other courses that the key length will give the strength of the access security proportionally. In our project information access and repository, elsewhere password is required, we shall have a strong enough security key that is of at least 8 combined sigh, alphabet, uppercase, lowercase, numbers, etc.

References

[1] H. Z. M. Singh S. Ahmed, "Nagini: Project management plan," University of South-Eastern Norway, Project management plan (PMP), Jun. 2, 2020.

[2] ——, "Nagini: Prototype documentation," University of South-Eastern Norway, Prototype Documentation, Jun. 2, 2020.

[3] ISO, *Systems and software engineering - system life cycle processes*, 1st ed., ISO 15288, International Organization for Standardization, Case postale 56, CH-1211 Geneva 20, May 2015, pp. 12, 13.

[4] Wikipedia contributors, *Nāga Wikipedia, the free encyclopedia*, [Online; accessed 31-May-2020], 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=N%5C%C4%5C%81ga&oldid=959532877.

[5] W. M. DeBusk, "Unmanned aerial vehicle systems for disaster relief: Tornado alley," 2009. DOI: https://ntrs.nasa.gov/search.jsp?R=20090036330.

[6] H. T̃. Arnold R.D Yamaguchi, "Search and rescue with autonomous flying robots through behavior-based cooperative intelligence," *Int J Humanitarian Action*, vol. 3, 2018. DOI: https://doi.org/10.1186/s41018-018-0045-4.

[7] SmartCitiesWorld. (Jun. 14, 2019). Autonomous drone trialled for emergency management, [Online]. Available: https://www.smartcitiesworld.net/news/news/autonomous-drone-trialled-for-emergency-management-4276.

[8] K. R. C. M. Erdelj E. Natalizio and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, 2017. DOI: https://doi.org/10.1109/MPRV.2017.11.

[9] K. Bonsor. (Apr. 19, 2001). How snakebots will work, [Online]. Available: https://electronics.howstuffworks.com/snakebot.htm.

[10] M. Singh, *Nagini: Master.ino*, 2020. [Online]. Available: https://github.com/Miniontab/Nagini/blob/master/Master.ino.

[11] J. T. G. Pål Liljebäck Øyvind Stavdahl, *Snake Robots: Modelling, Mechatronics, and Control*, 1st ed., ser. Advances in Industrial Control. Springer-Verlag London, 2013, pp. 40–42, ISBN: 978-1-4471-2995-0,978-1-4471-2996-7.

[12] M. Shead. (Feb. 11, 2018). Understanding state machines, [Online]. Available: https://www.freecodecamp.org/news/state-machines-basics-of-computer-science-d42855debc66/.

[13] M. F. Ben-Ari M., *Elements of Robotics*. Springer, Cham, Oct. 27, 2017, ch. 4, pp. 55–61, ISBN: 978-3-319-62533-1. DOI: https://doi.org/10.1007/978-3-319-62533-1_4.

[14] S. P.R., *A Practical Introduction to Hardware/Software Codesign*. Springer, Boston, MA, Oct. 6, 2012, ch. 5, pp. 125–132, ISBN: 978-1-4614-3737-6. DOI: https://doi.org/10.1007/978-1-4614-3737-6_5.

[15] V. Paradigm. (). What is work breakdown structure? [Online]. Available: https://www.visual-paradigm.com/guide/project-management/what-is-work-breakdown-structure/.

[16] F. Brotherton S. A. (Oct. 19, 2008). Applying the work breakdown structure to the project management lifecycle, [Online]. Available: https://www.pmi.org/learning/library/applying-work-breakdown-structure-project-lifecycle-6979.

[17] ISO, *Systems and software engineering - system life cycle processes*, 1st ed., ISO 15288, International Organization for Standardization, Case postale 56, CH-1211 Geneva 20, May 2015.

[18] S. Ray. (Nov. 19, 2019). Understanding critical path in project management, [Online]. Available: https://www.projectmanager.com/blog/understanding-critical-path-project-management.

[19] A. Slate. (Oct. 25, 2018). Critical path method: A project management essential, [Online]. Available: https://www.wrike.com/blog/critical-path-is-easy-as-123/.

[20] J. Andersson. (Jan. 20, 2020). Activities and life cycle models, [Online]. Available: https://www.dropbox.com/s/gc4akzbwtp9nu8y/T05.1%5C%20-%5C%20Activities%5C%20and%5C%20lifecycle%5C%20models.pdf?dl=0.

[21] pixycam. (Feb. 18, 2019). Pixy2 overview, [Online]. Available: https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:overview.

[22] B. Boehm and A. Jain, "A value-based theory of systems engineering," *INCOSE International Symposium*, vol. 16, Jul. 2006. DOI: 10.1002/j.2334-5837.2006.tb02785.x.

[23] D. A. U. PRESS, *Systems engineering fundamentals*, IEE TR 24748-1, Department of Defense: Systems Management College, FORT BELVOIR, VIRGINIA, Jan. 2001. [Online]. Available: http://acqnotes.com/wp-content/uploads/2017/07/DAU-Systems-Engineering-Fundamentals.pdf.

[24] J. C. of Staff DOD, *Department of Defense Dictionary of Military and Associated Terms.* DIANE Publishing, 2004, ISBN: 1428980458,9781428980457. [Online]. Available: http://gen.lib.rus.ec/book/index.php?md5=cf8c5b9fa1958e5906ecf9bc2c4e68a4.

[25] D. Knuth. (). Knuth: Computers and typesetting, [Online]. Available: https://www.afconsult.com/en/do-business/our-services/technology/defence/integrated-logistics-support/.

[26] A. Sols, *Systems Engineering: Theory and Practice.* Universidad Pontificia Comillas, 2000, ISBN: 9788484685395. [Online]. Available: https://books.google.no/books?id=ssH1oQEACAAJ.

[27] ISO, *Iso/iec/ieee international standard – systems and software engineering - content of life-cycle information items (documentation)*, 1st ed., International Organization for Standardization, Case postale 56, CH-1211 Geneva 20, May 2019.

[28] N. A. R. Tariq M. Rahim, "Dronaid : A smart human detection drone for rescue," in *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT and IoT (HONET-ICT)*, 2018. DOI: https://doi.org/10.1109/HONET.2018.8551326.

[29] M. Mori and S. Hirose, "Development of active cord mechanism acm-r3 with agile 3d mobility," vol. 3, Feb. 2001, 1552–1557 vol.3, ISBN: 0-7803-6612-3. DOI: 10.1109/IROS.2001.977200.

[30] S. L. Sohn. (Nov. 24, 2019). Progressive elaboration, [Online]. Available: https://www.projectmanagement.com/wikis/295452/Progressive-Elaboration.

[31] ISO, *Systems and software engineering - life cycle processes - project management*, 1st ed., ISO 16326, International Organization for Standardization, Case postale 56, CH-1211 Geneva 20, Dec. 2009.

[32] P. M. Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 3rd, ser. PMBOK Guides. Project Management Institute, 2004, ISBN: 9781930699458.

[33] ISO, *Guidance on project management*, 1st ed., ISO 21500, International Organization for Standardization, Case postale 56, CH-1211 Geneva 20, Sep. 2012.

[34] ——, *Systems and software engineering — life cycle management — part 4: Systems engineering planning*, 1st ed., ISO 24748-4, International Organization for Standardization, Case postale 56, CH-1211 Geneva 20, May 2016.

[35] C. Petersen, *The Practical Guide to Project Management*, 1st ed. bookboon, 2013, ISBN: 978-87-403-0524-1.

[36] IEE, *Systems and software engineering—life cycle management—part 1: Guide for life cycle management*, 1st ed., IEE TR 24748-1, IEE Computer Society, 3 Park Avenue, NY 10016-5997, Jun. 2011.

[37] ISO, *Systems and software engineering – software life cycle processes*, 2017, pp. 1–157.

[38] Wikipedia contributors, *Scrum (software development) — Wikipedia, the free encyclopedia*, [Online; accessed 29-May-2020], 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Scrum_(software_development)&oldid=959495566.

[39] J. Andersson. (Jan. 20, 2020). Extra network planning, [Online]. Available: https://www.dropbox.com/s/t7api1e42irtcpo/T05.2%5C%20-%5C%20Extra%5C%20Network%5C%20planning.pdf?dl=0.

[40] A. Shenhar and D. Dvir, *Reinventing Project Management : The Diamond Approach To Successful Growth And Innovation.* Harvard Business Review Press, 2007, ISBN: 9781591398004. [Online]. Available: http://ezproxy2.usn.no:2056/login.aspx?direct=true&db=nlebk&AN=675086&site=ehost-live.

[41] D. Workshop. (Feb. 10, 2018). Stepper motors with arduino - controlling bipolar & unipolar stepper motors, [Online]. Available: https://youtu.be/0qwrnUeSpYQ.

[42] ——, (Jul. 14, 2018). Getting started with lidar, [Online]. Available: https://youtu.be/VhbFbxyOI1k.

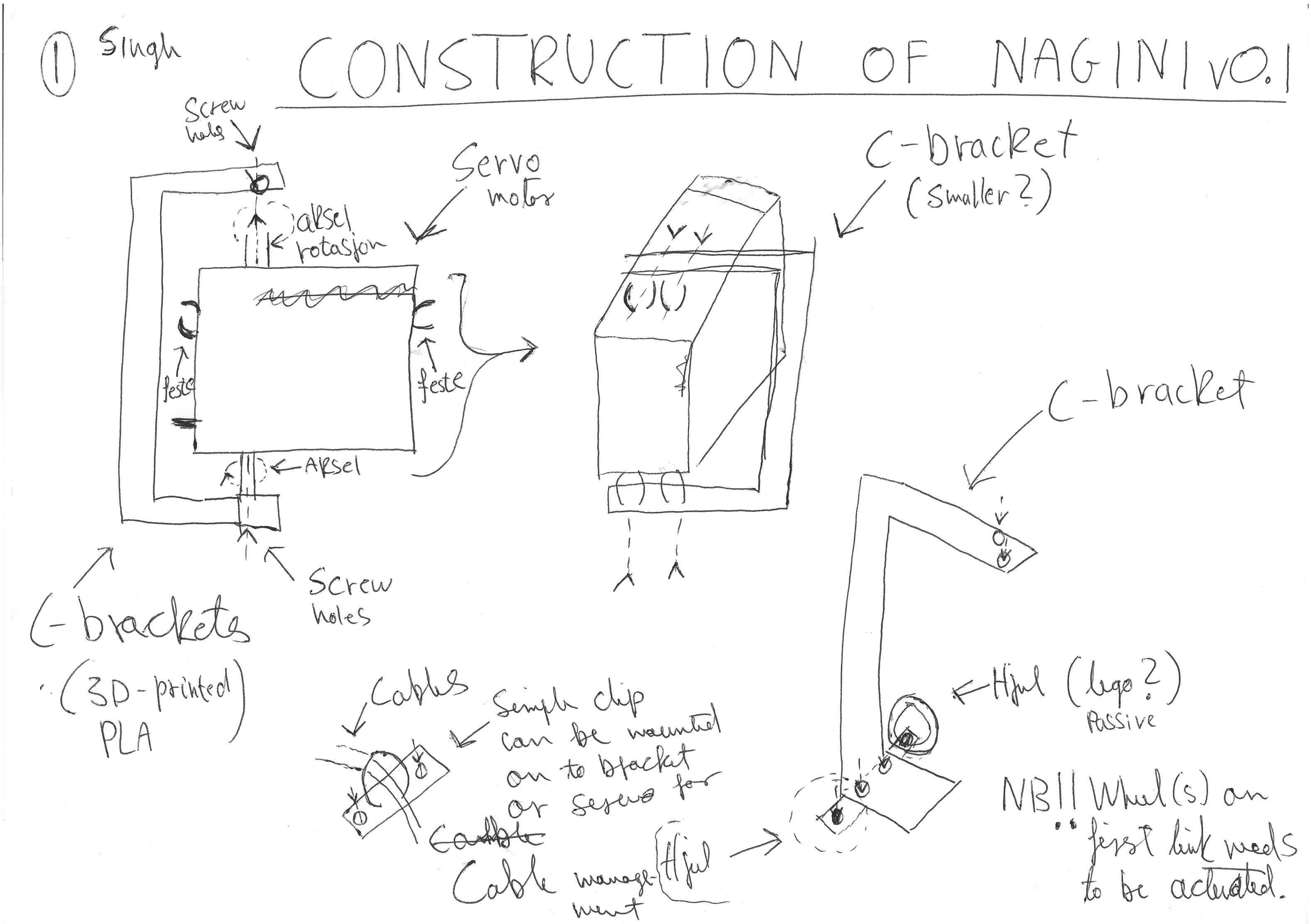
[43] P. McWhorter. (May 31, 2019). Arduino tutorial 1: Setting up and programming the arduino for absolute beginners, [Online]. Available: https://youtu.be/fJWR7dBuc18.

[44] Scrum.org. (May 30, 2020). What is scrum? [Online]. Available: https://www.scrum.org/resources/what-is-scrum.

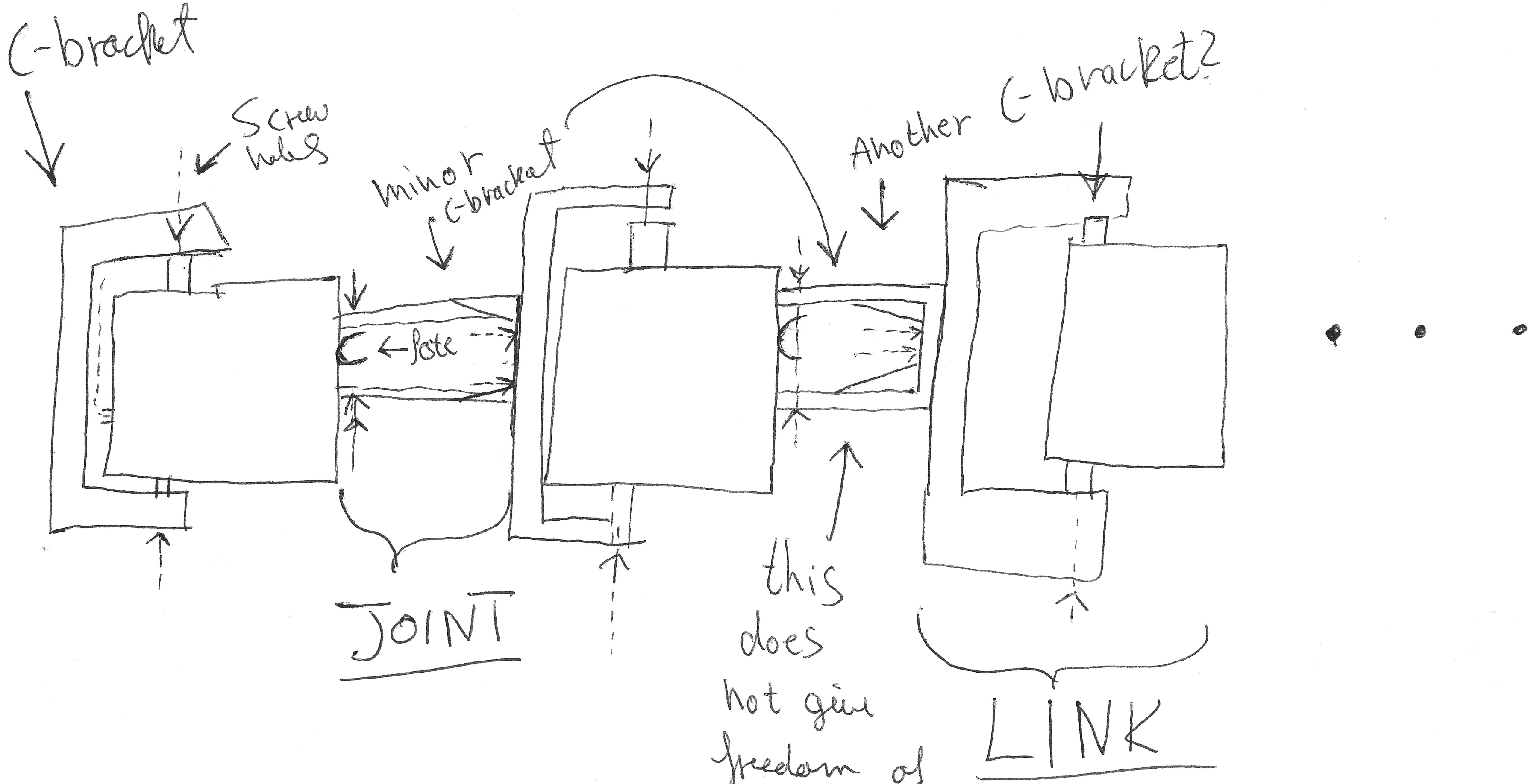
[45] K. Schwaber and J. Sutherland. (2017). The scrum guide™, [Online]. Available: https://www.scrumguides.org/scrum-guide.html.

Appendices

Appendix A Construction plan of Nagini v0.1



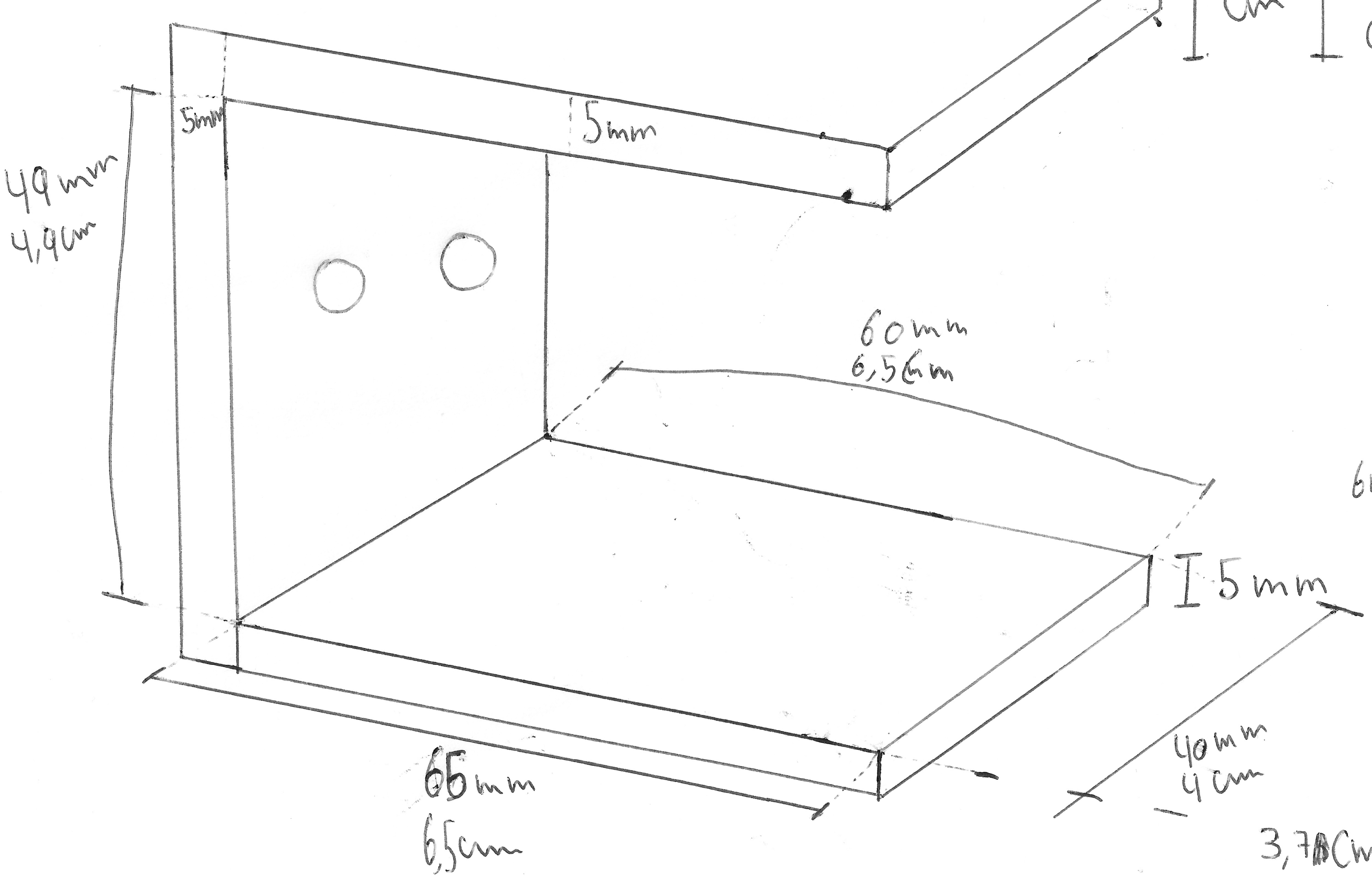
(2) Single ~~TOTAL~~ LENGTH

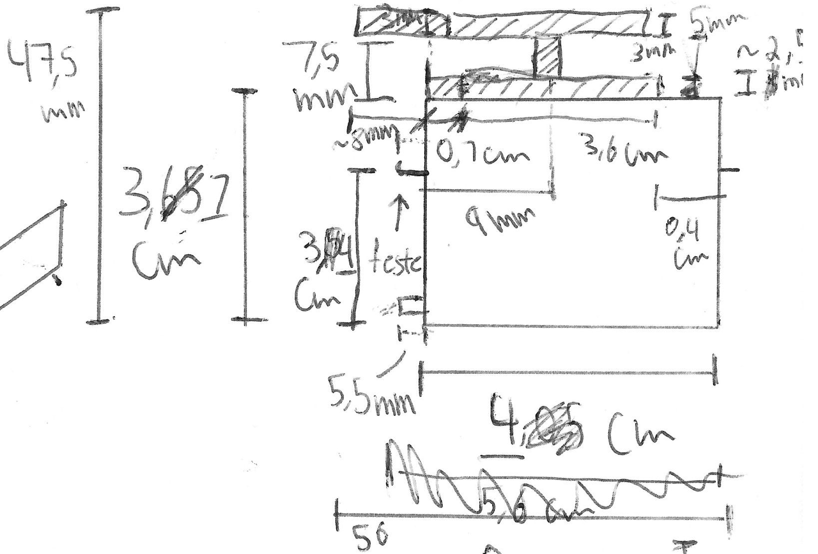


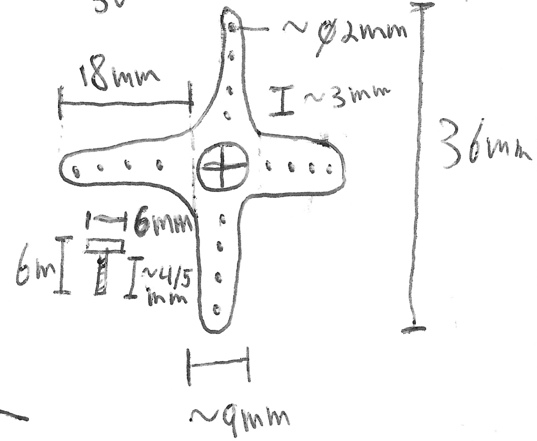
this does not give freedom of movement horizontally.

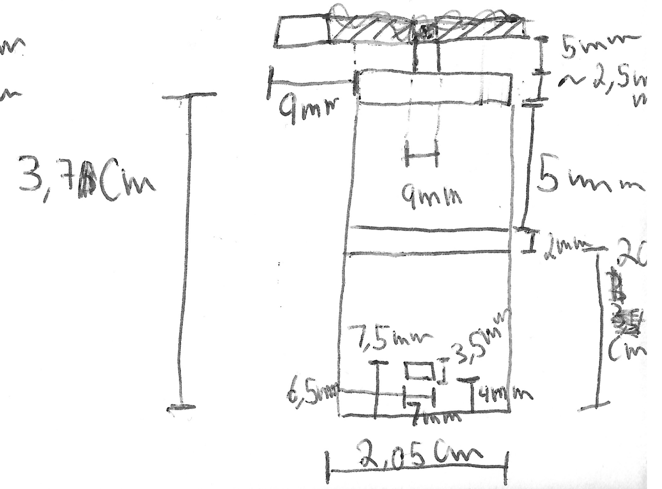
2A) C-brackets dimensions and Servomotor dimensions

Singh





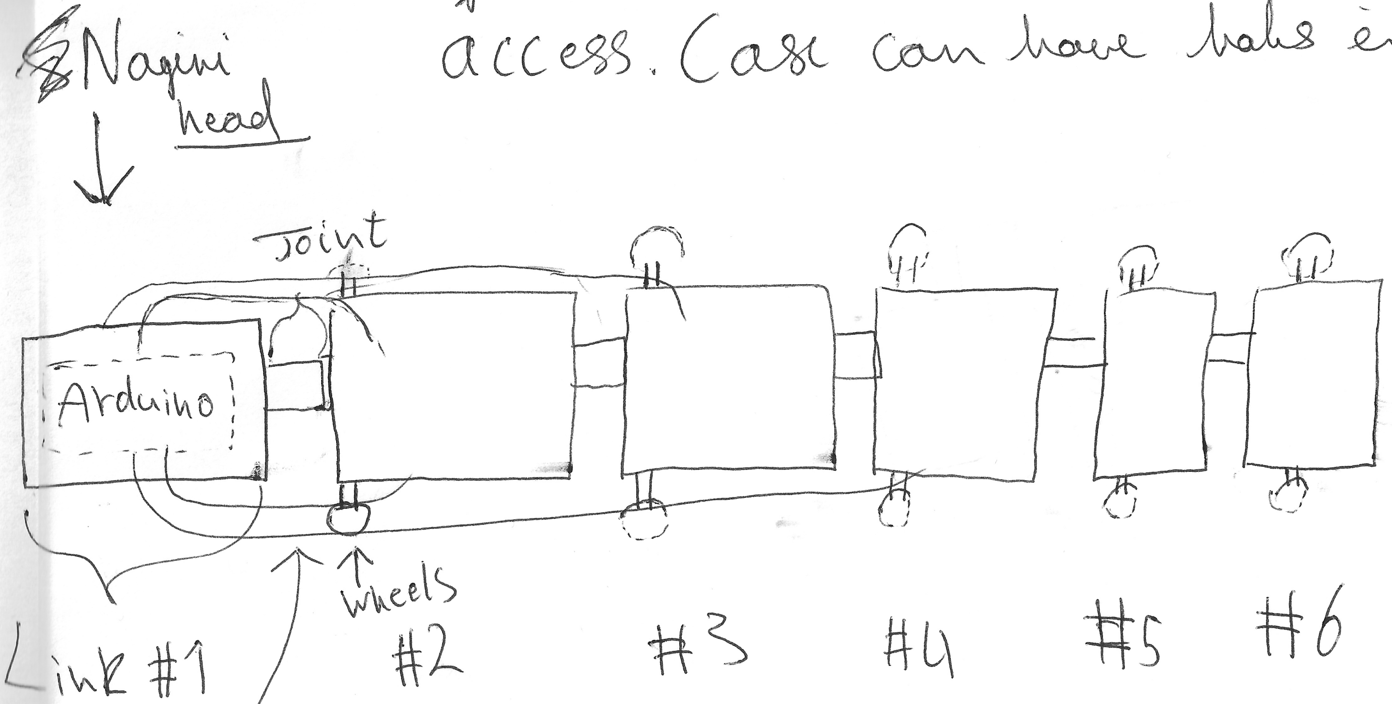




3 Singh

Top-View of Nagini

Arduino can be ~~w~~ velcro'ed ontop or in link #1 with a easy 3D-printed case on-top for protection and access. Case can have holes in it for wire management.

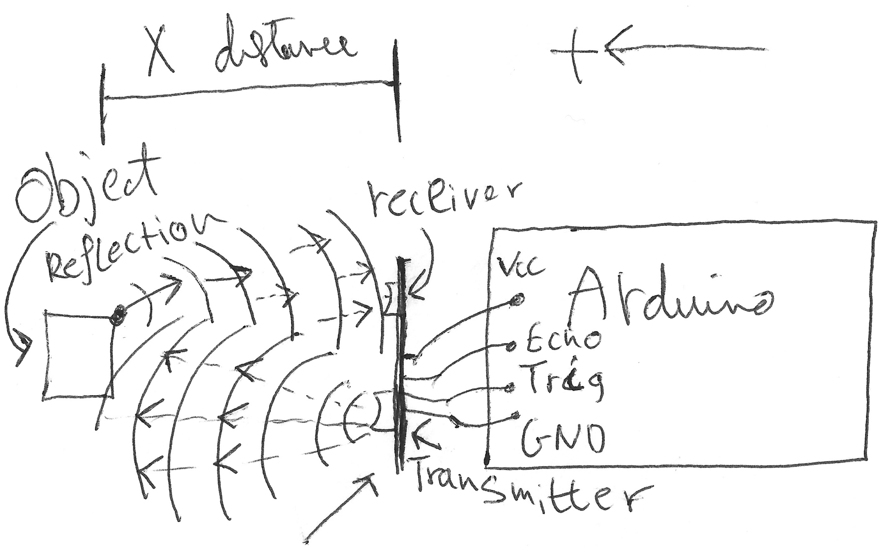


Wires can go ontop of the model through clips (page 7).

4 Singh

Detection Subsystem

A

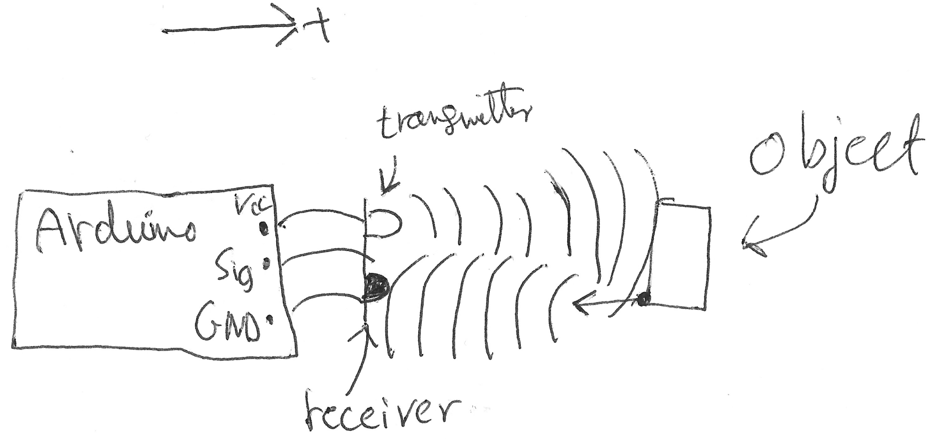


Ultrasonic Sensor HC-SR04

Time between transmission and reception of signal can help us calculate distance ~~between~~ to object, because we know V_{sound} in air.

To follow the object, the robot has to travel X-distance

B



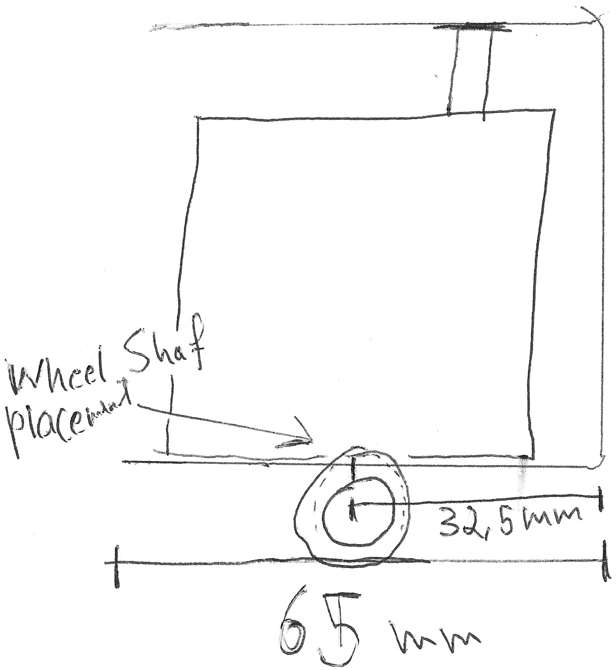
IR Obstacle avoidance sensor

The IR transmitter sends IR light, and the receiver gets the reflected signal back and understands that there's an object there.

5 3.3.2020
Seingh and Samuel

mo

Wheel- and shaft placement

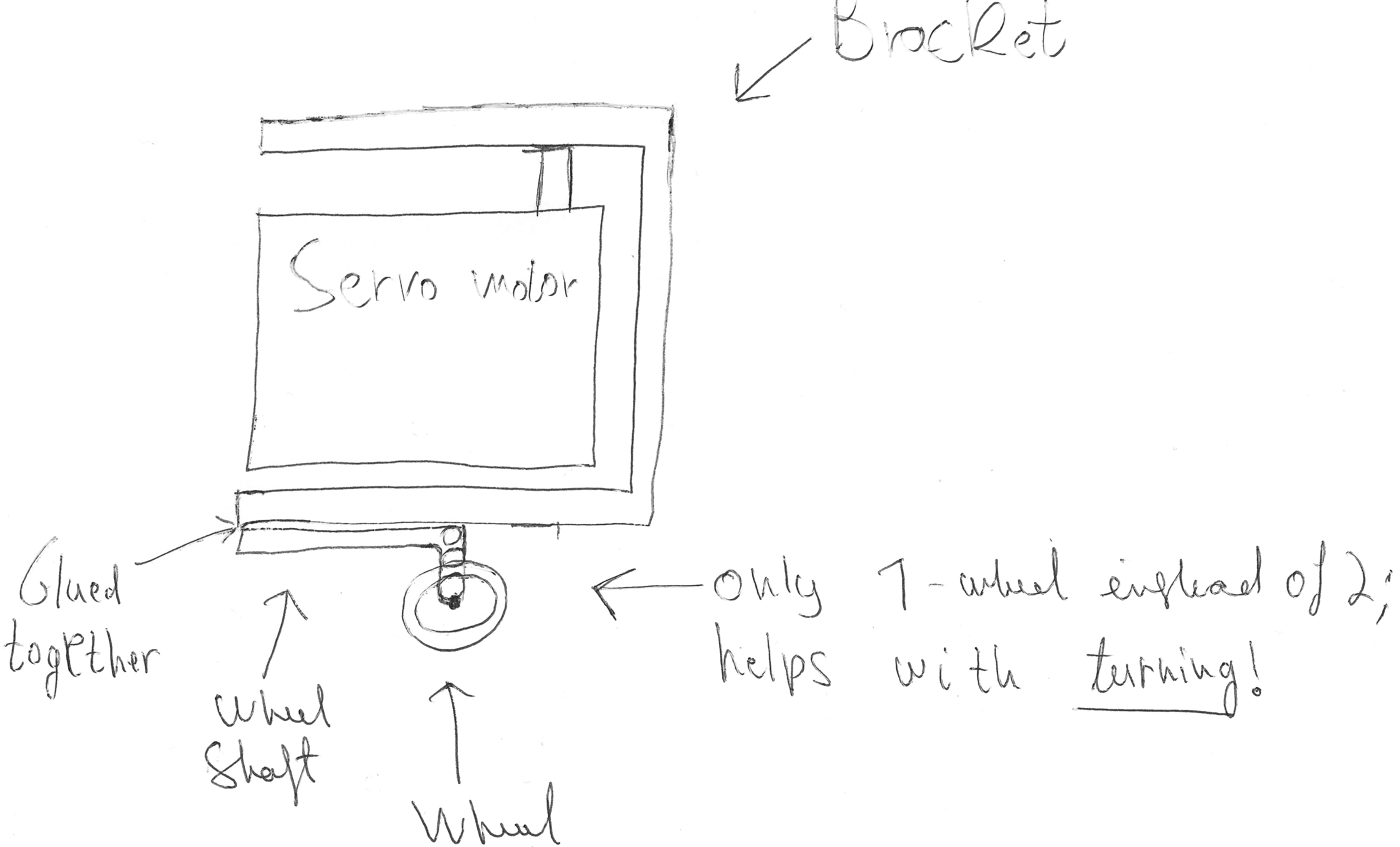


Problem! When servo rotates 90° left or right, it hits the wheels!

See 6 for alternate wheel solution!

6 03.03.2020
Singh and Samuel

Re-did wheels because 1. pair of wheels were obstructive.



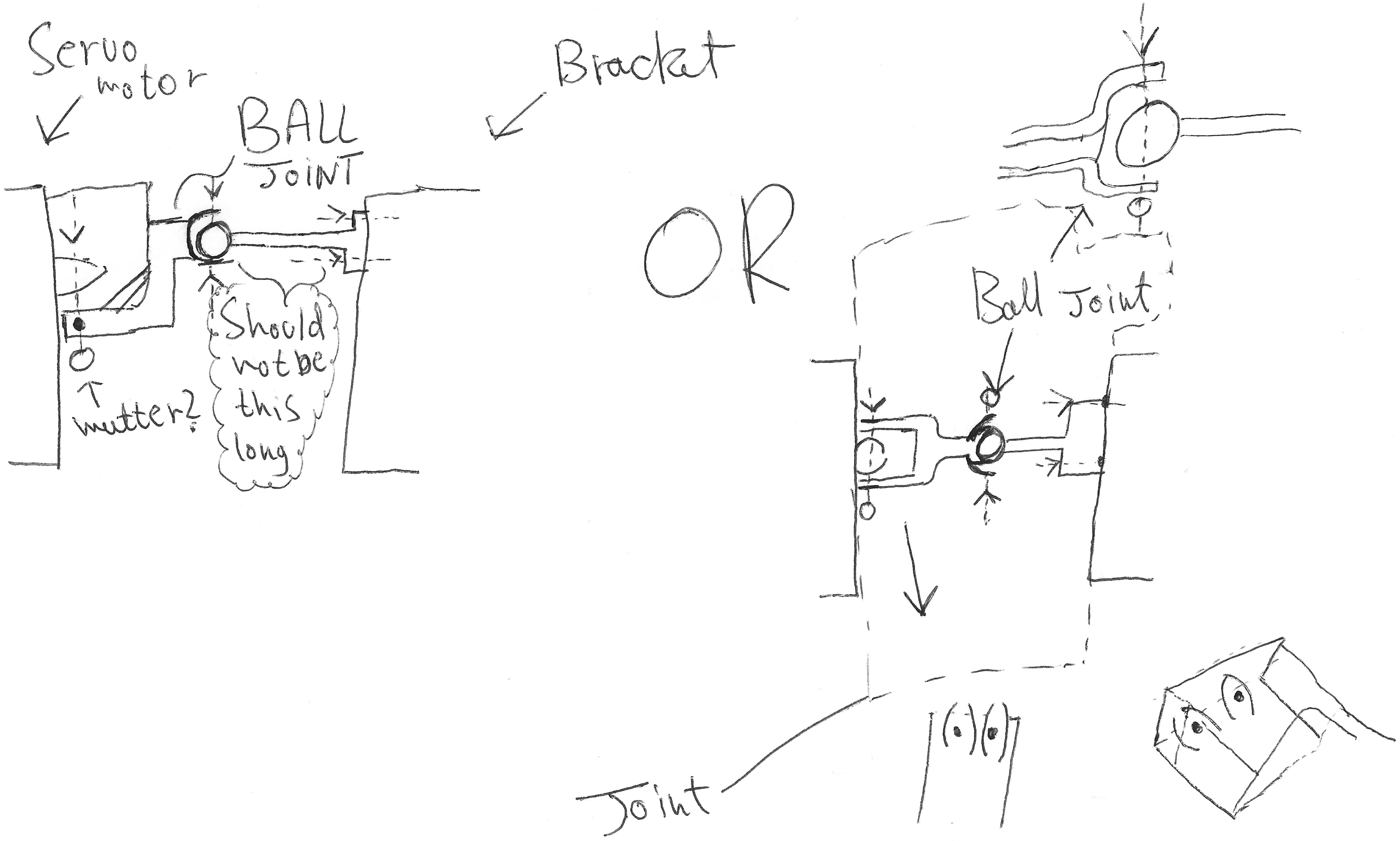
Singh

Things to do - 03.03.2020 17:12

- Harry manufacturing of prototyping parts since Richard isn't available rest of week 10.
  - Laser cut rest of prototyping parts ✓
    - Joint
    - Link
  - Build prototype Nagini with laser cut parts.
- ~~Design Joint - 3D~~
- Design snake head with room for Arduino ~~and~~ (batteries?)
- 3D-print ALL parts when Richard is back

7 Singh

Newtype of Joint (Ball Joint)

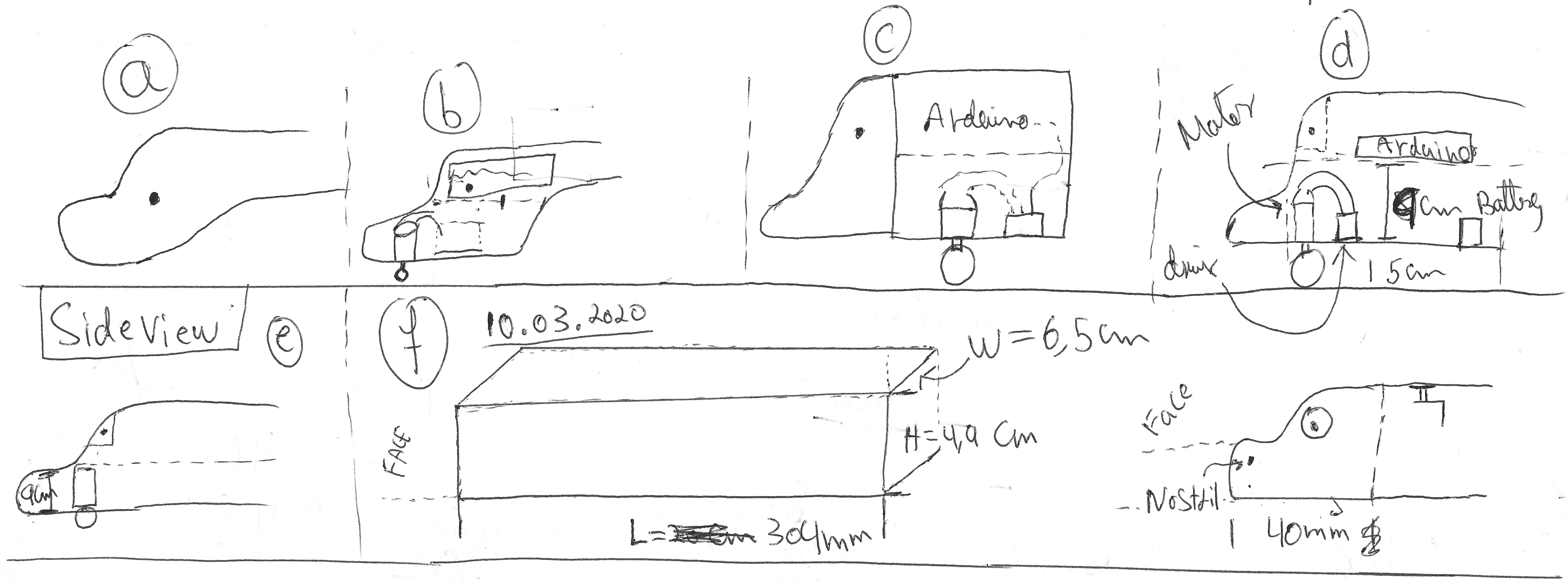


This new joint ~~is~~ grants freedom of movement to link$_{n+1}$, ~~&~~ unlike Joint V.1.

09.03.2020

Snake Head Model

Samuel and Singh

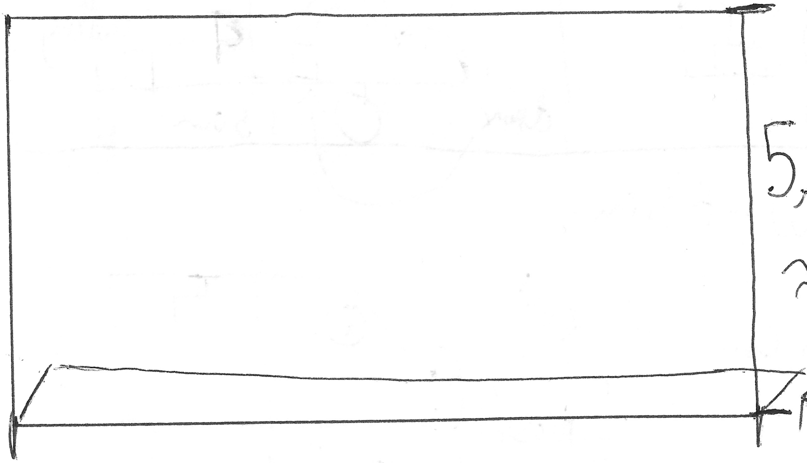


L = ~~19,5~~ ~~20 Cm~~, W = 6,5 cm, H = Same as bracket, 4,9cm

8a Singh and Samuel

Measurements of various components

Arduino measurement



5,3cm

≈ 5,5cm = W

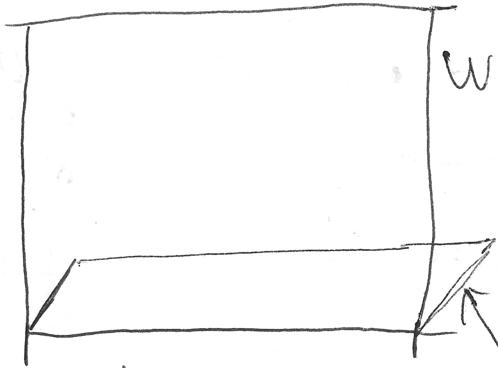
h = 1,8cm

≈ 2cm

3,5cm w/cable

l = 10,8cm ≈ 11,00cm

Motor driver measurements



W = 4,4cm

≈ 4,5cm

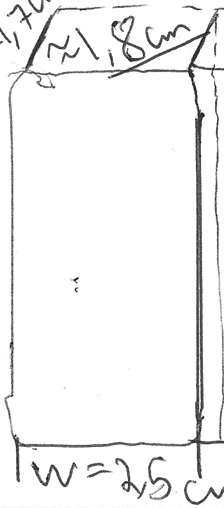
h = 2,9 cm

≈ 3,1 cm

l = 4,4cm

≈ 4,5cm

Battery

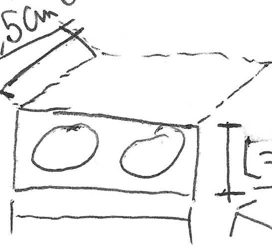


D = 1,7cm ≈ 1,8cm

h = 5cm

W = 2,5cm ≈ 2,7cm

US Sensor

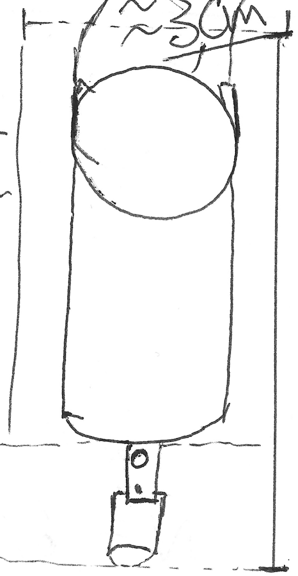


D = 3,5cm

L = 2,1cm

W = 4,7cm

DC-Motor mst.



W = 2,7cm

≈ 3cm

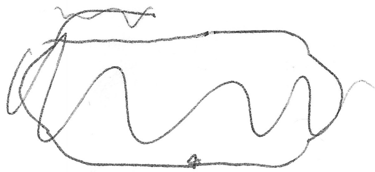
9cm

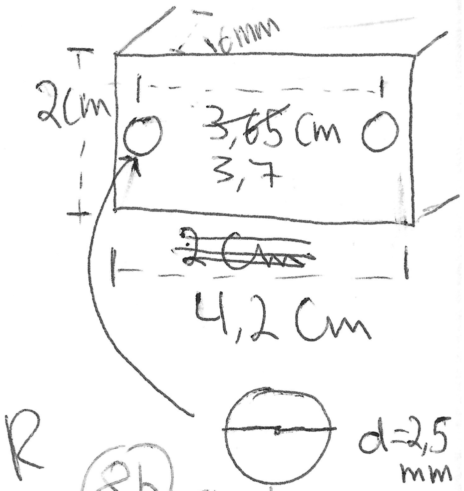
8cm

h = 11cm

≈ 12cm

for wire legspace





MINI LIDAR

8b Singh

10.03.2020

Singh

1. Snake head → 3D print!

2. Lower part of snake head with movement subsystem

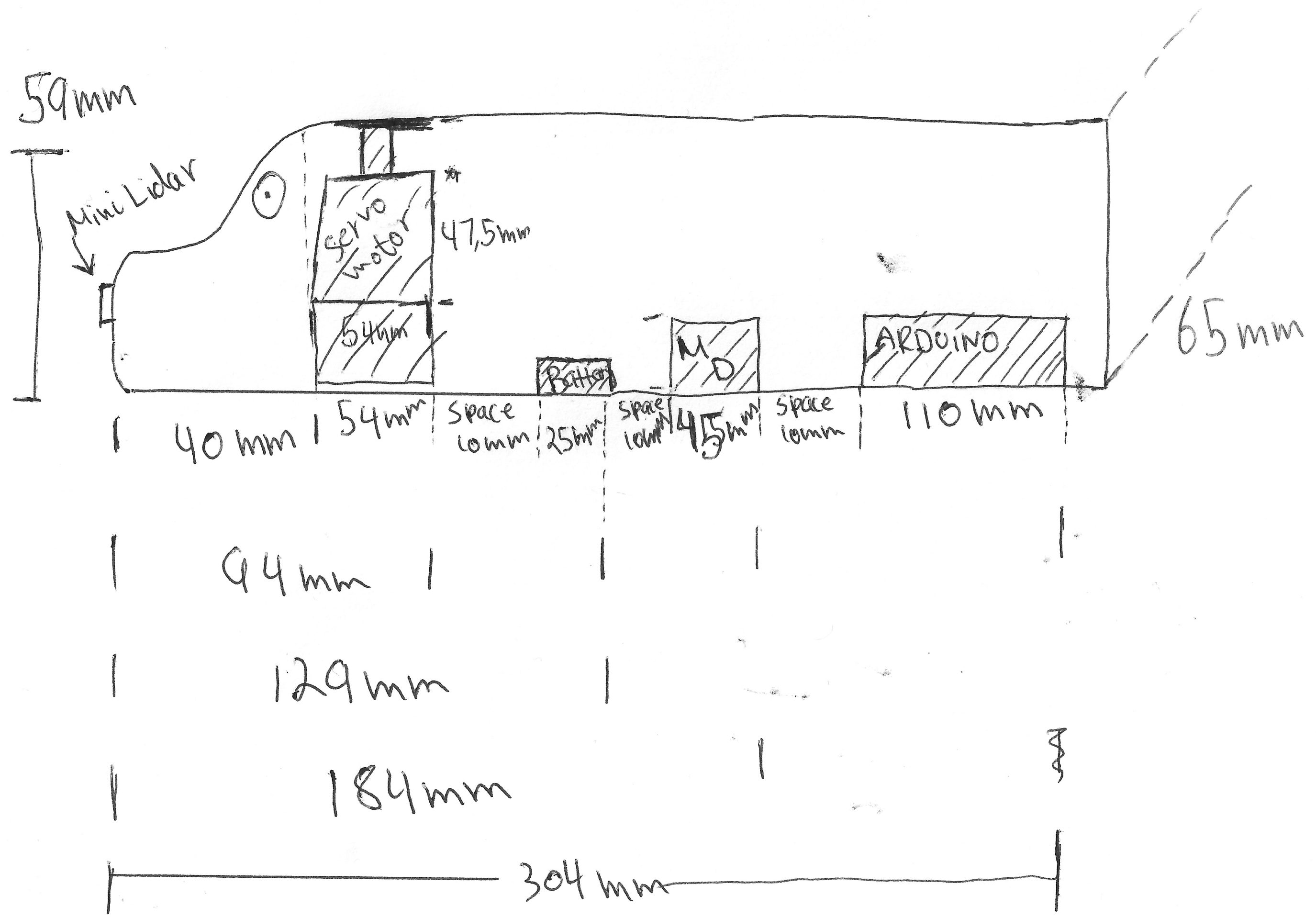
3. Connect everything

10.~~8~~03.2020

9

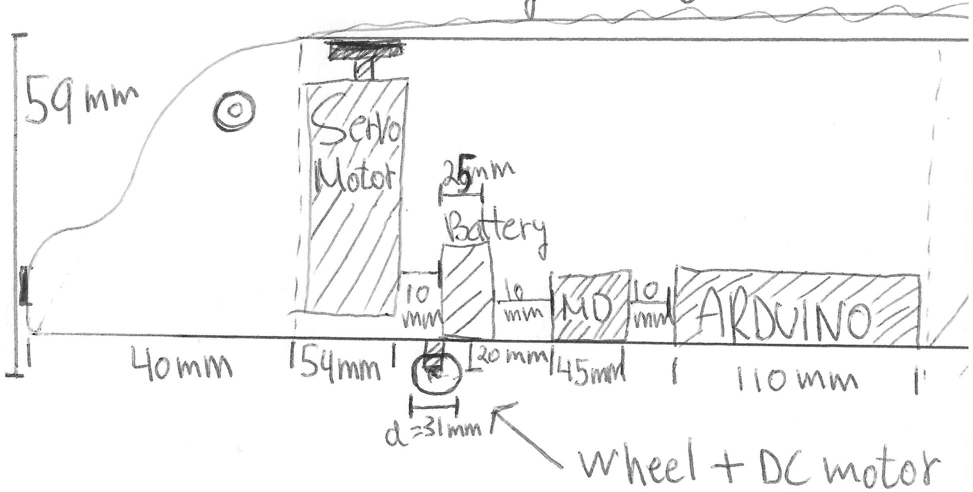
Snake head W/ components

Singh and Samuel



9a Singh

Snake height = 59mm
width = 65mm
length = 304mm



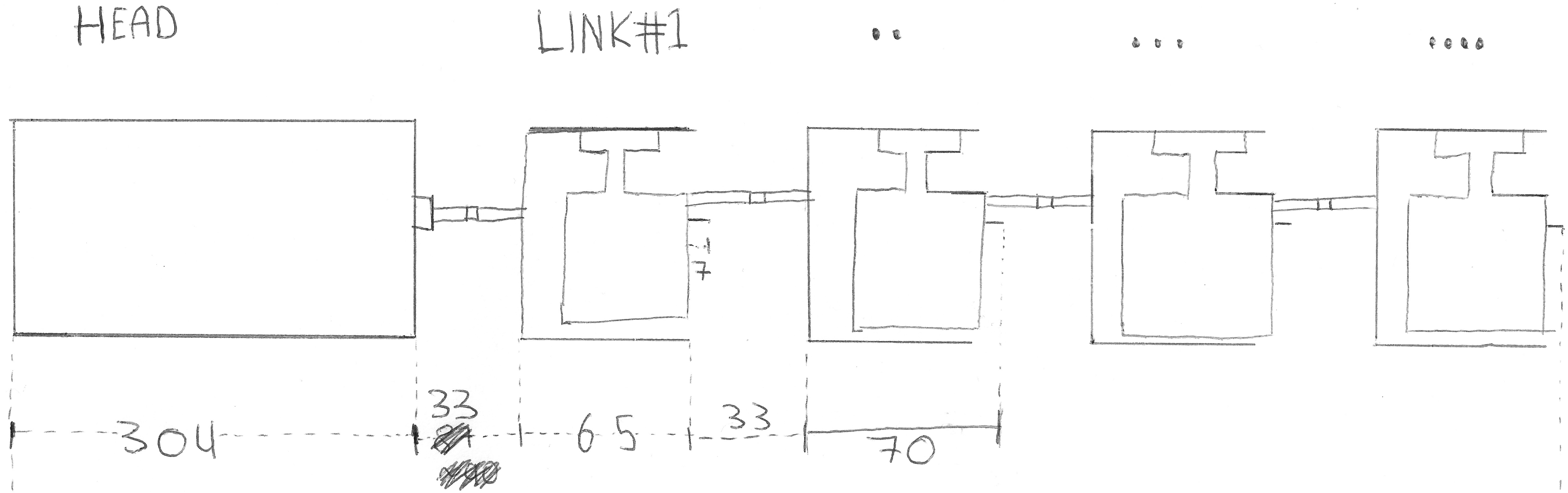
30~~0~~4mm

304 mm

Fri 13.03.2020 10

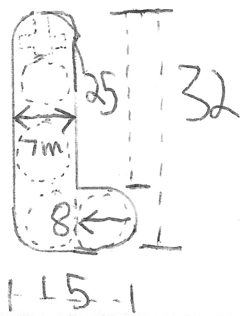
Overview of Nagini 0.1V

Singh

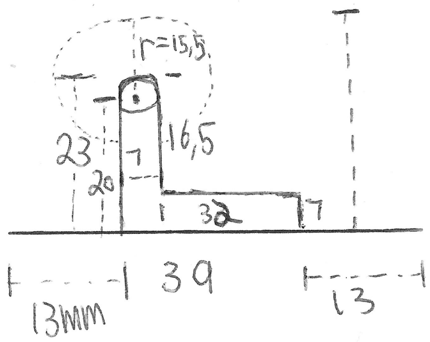


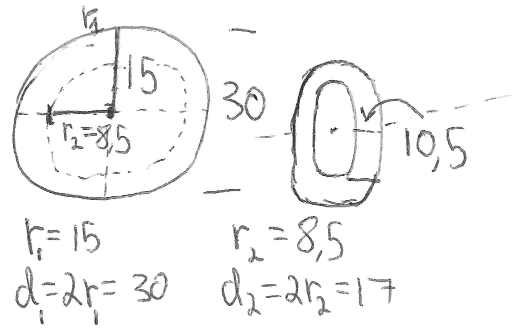
~~924 mm~~ 696mm

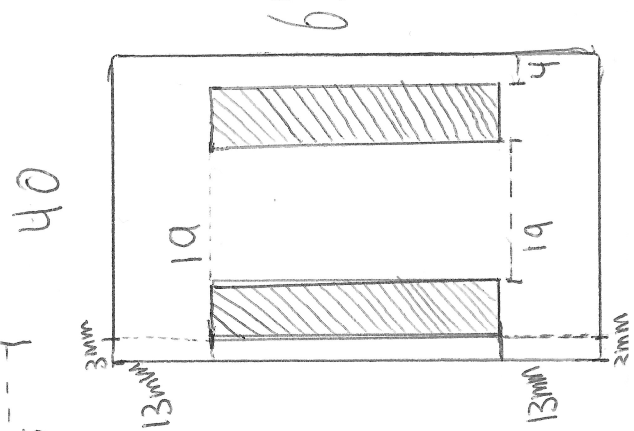
~~92,4 cm~~ 69,6 cm

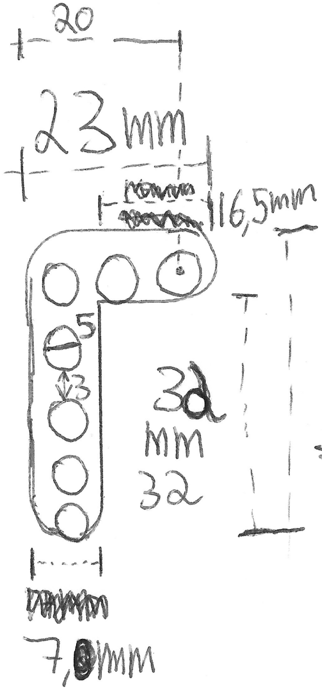
~~~1m~~ ~ 70 cm

0,7m


~~~

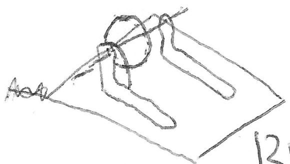
ALL UNITS
IN mm = 10·cm

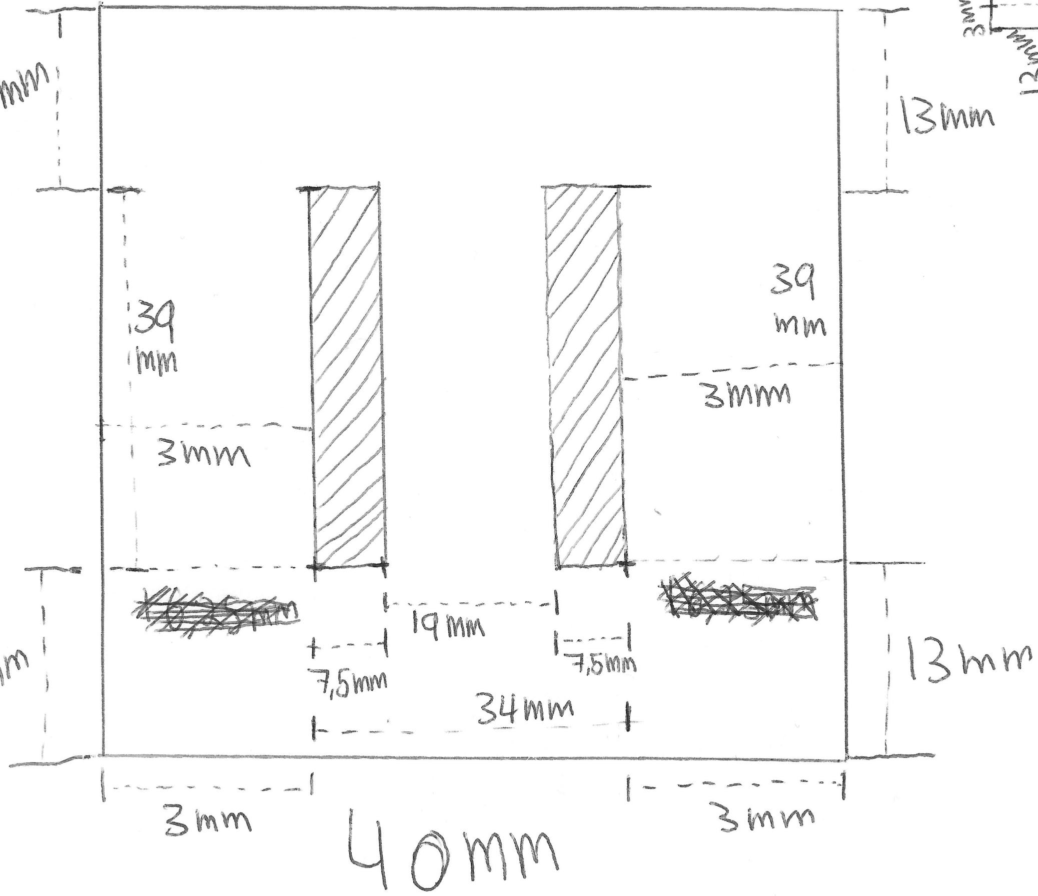




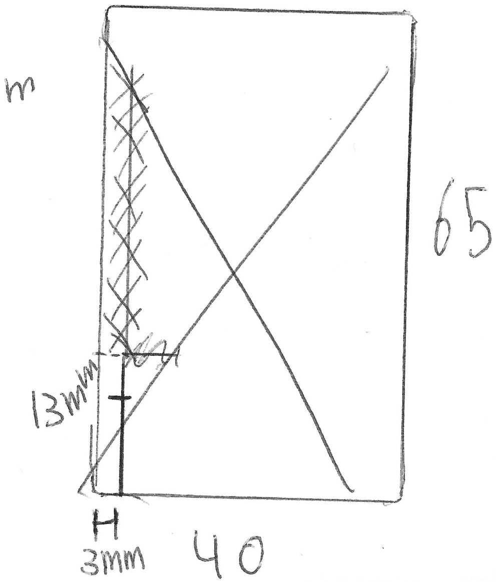








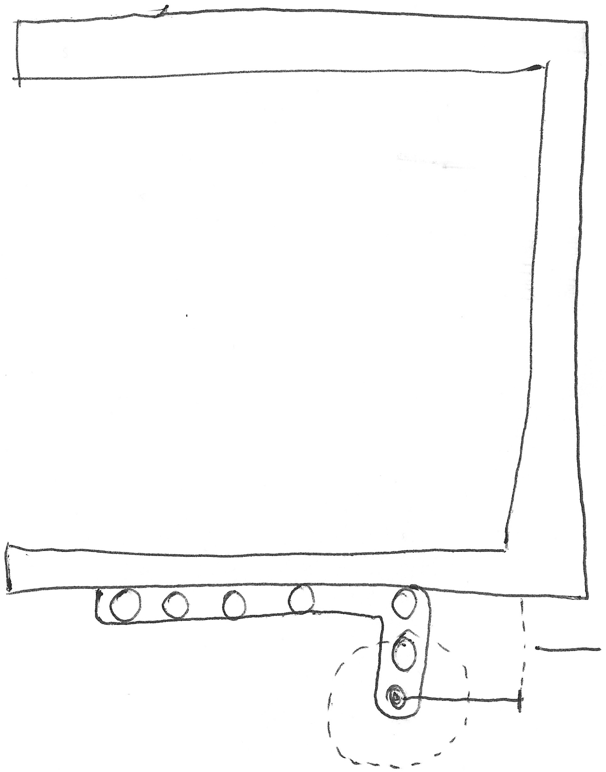
Bottom view of C-bracket



Wheels and
axle placement

SINGH
14.03.20 02:45 (11)

11a Singh

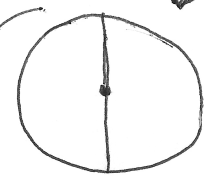


2cm

Wheel

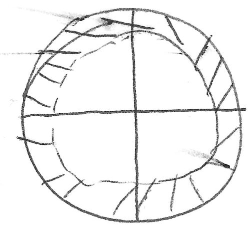
d = 2,5cm

~~2,Cm~~



Wheels Singh and Samuel

11b



d = 3,1 Cm

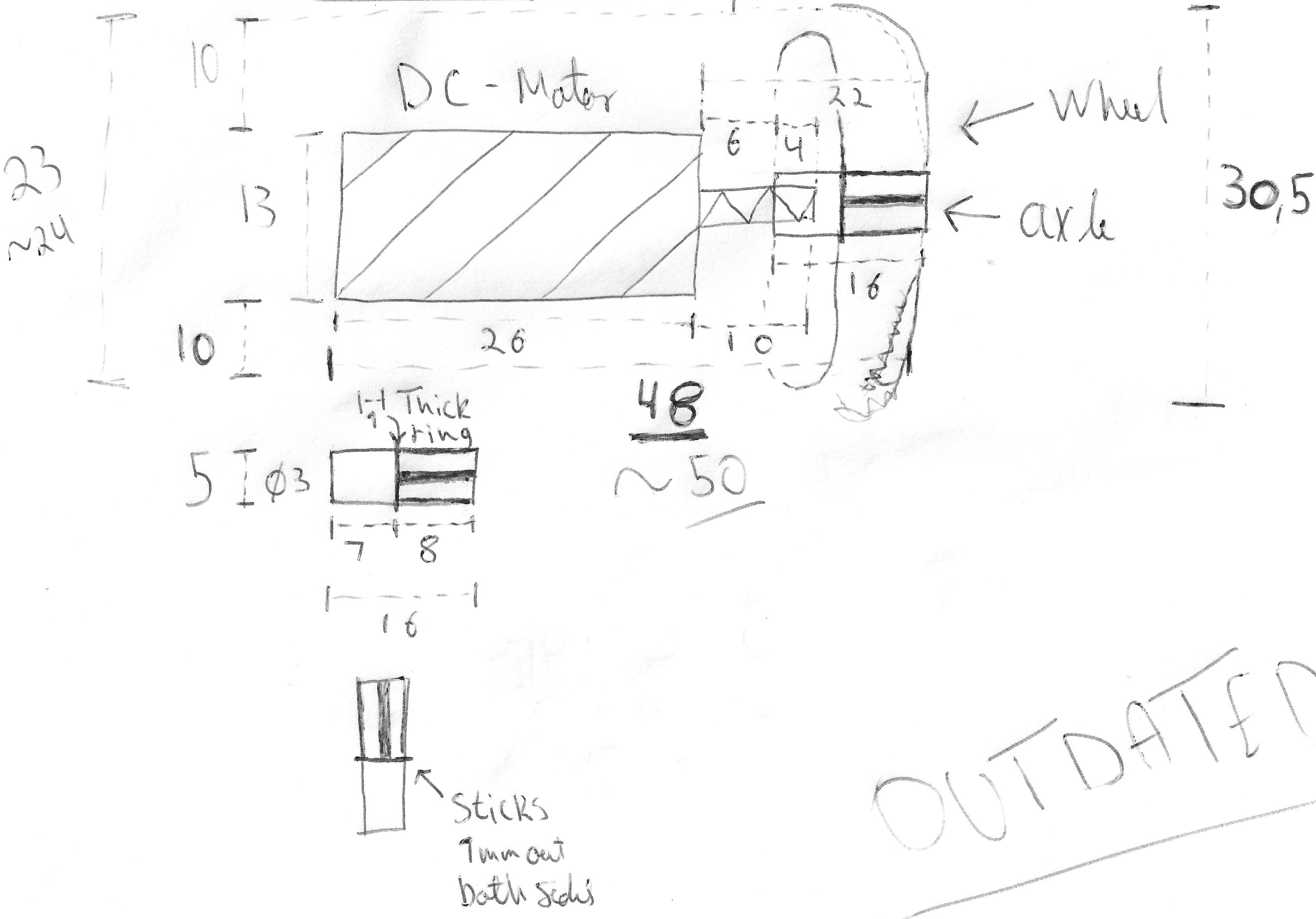
r = d/2 = 1,55 Cm

16.03.2020 Singh

12

Singh

DC-Motor w/ axle and wheel



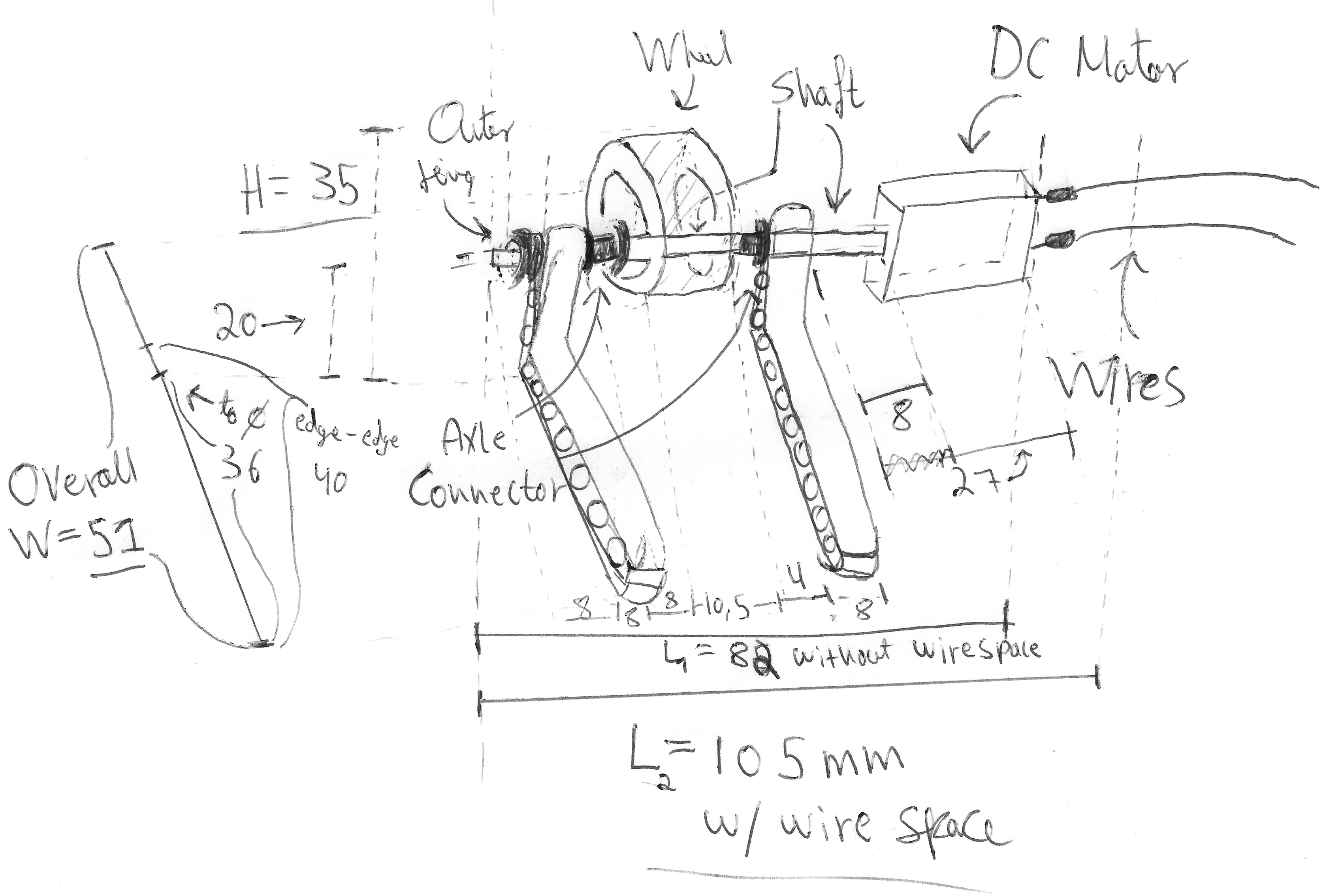
OUTDATED

17.03.2020 Sigh

ALL UNITS [mm]

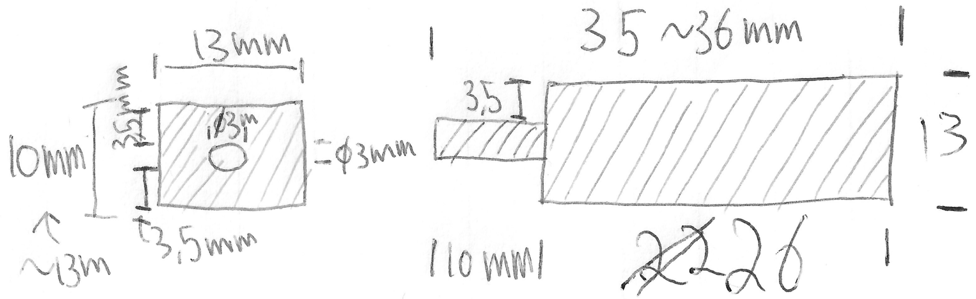
13

Movement Subsystem - DC motor w/ wheel



13a Singh

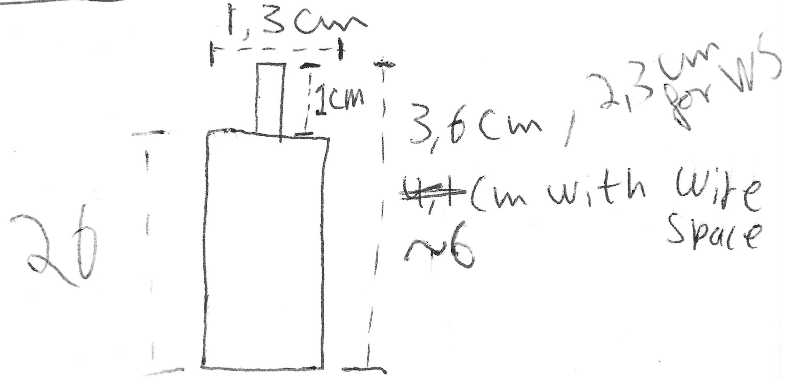
DC-MOTOR



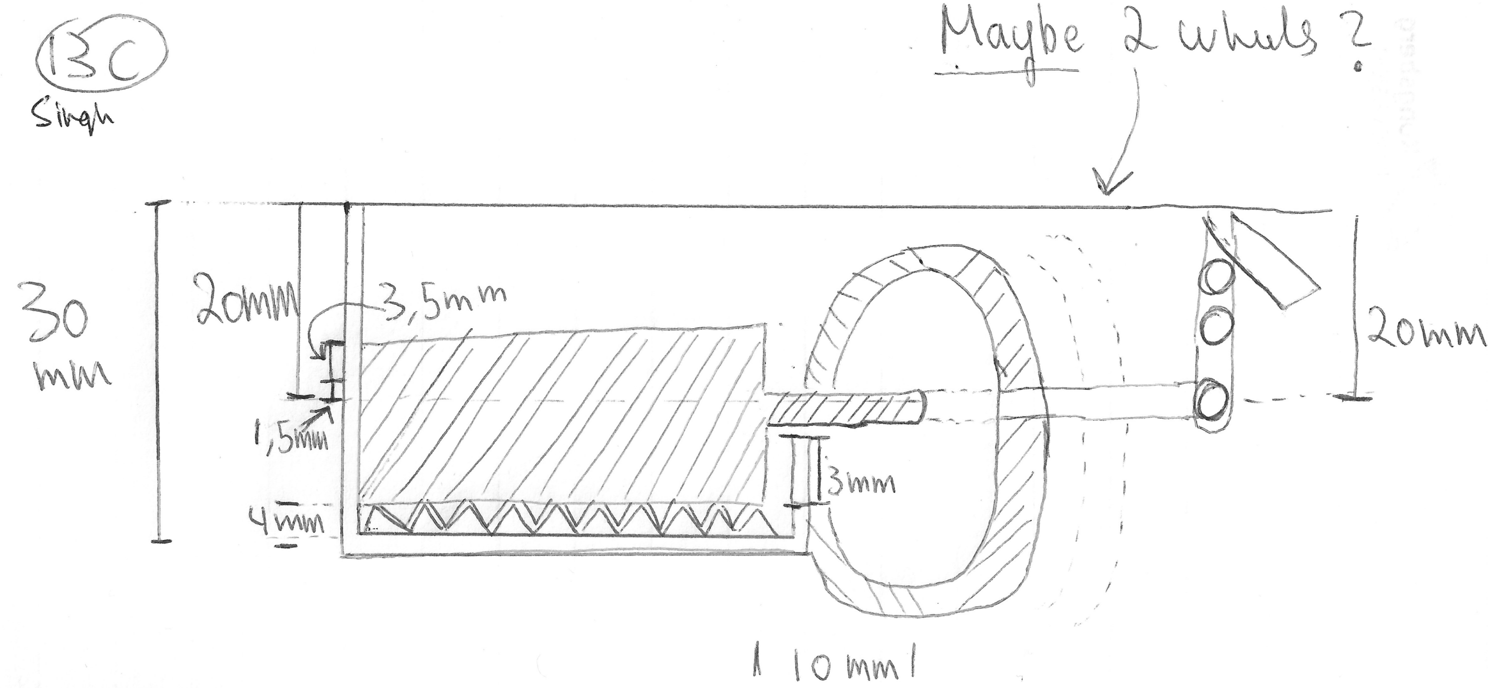
The motor is near ~~qudr~~ quadratic!

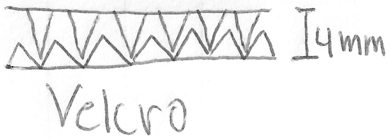
Singh 13b

~~Servo~~ DC Motor

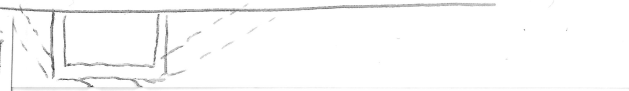


13C
Singh



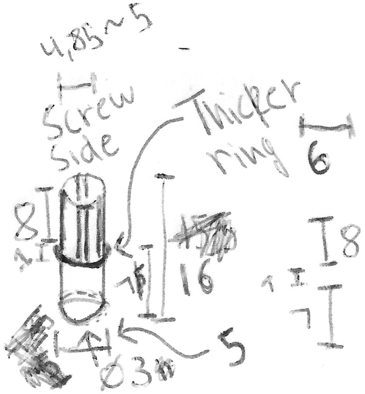


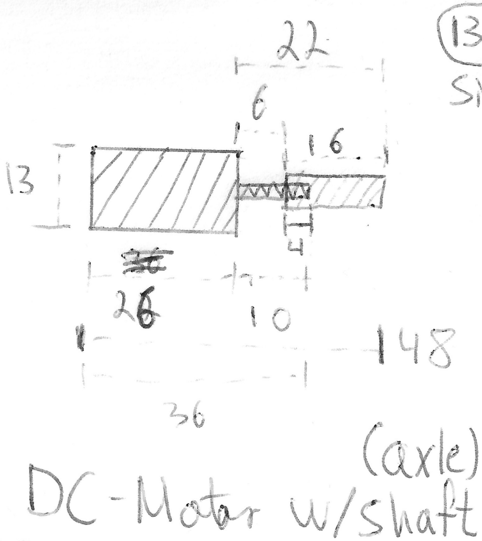
The water compartment will need support to baseline on both side?



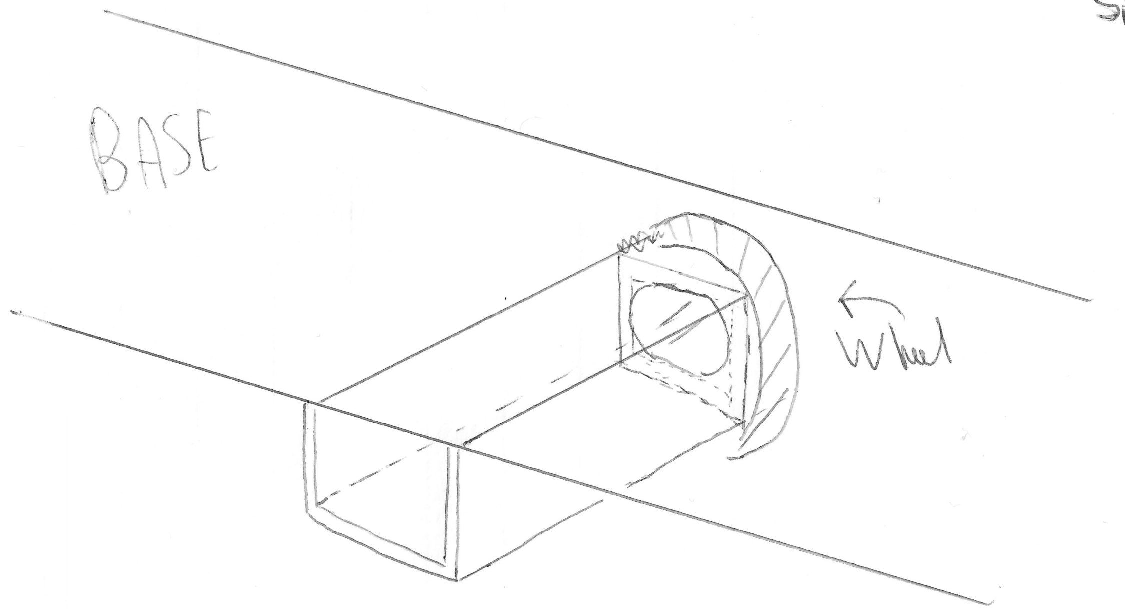
ALL UNITS IN mm.

13C.1
Singh

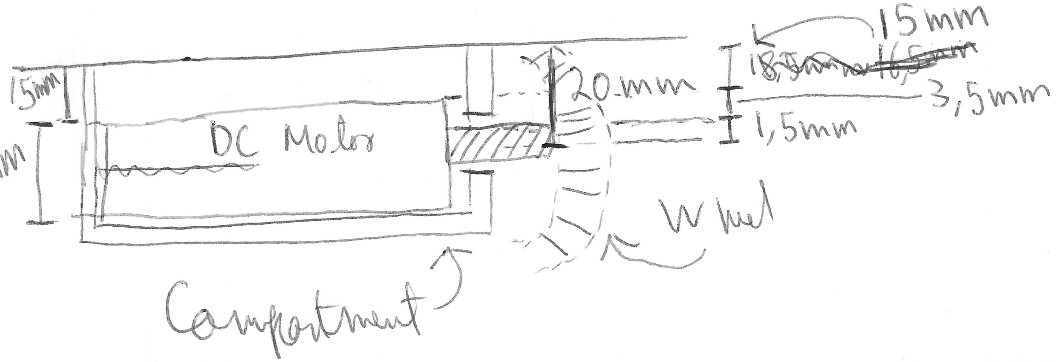




Single 13d

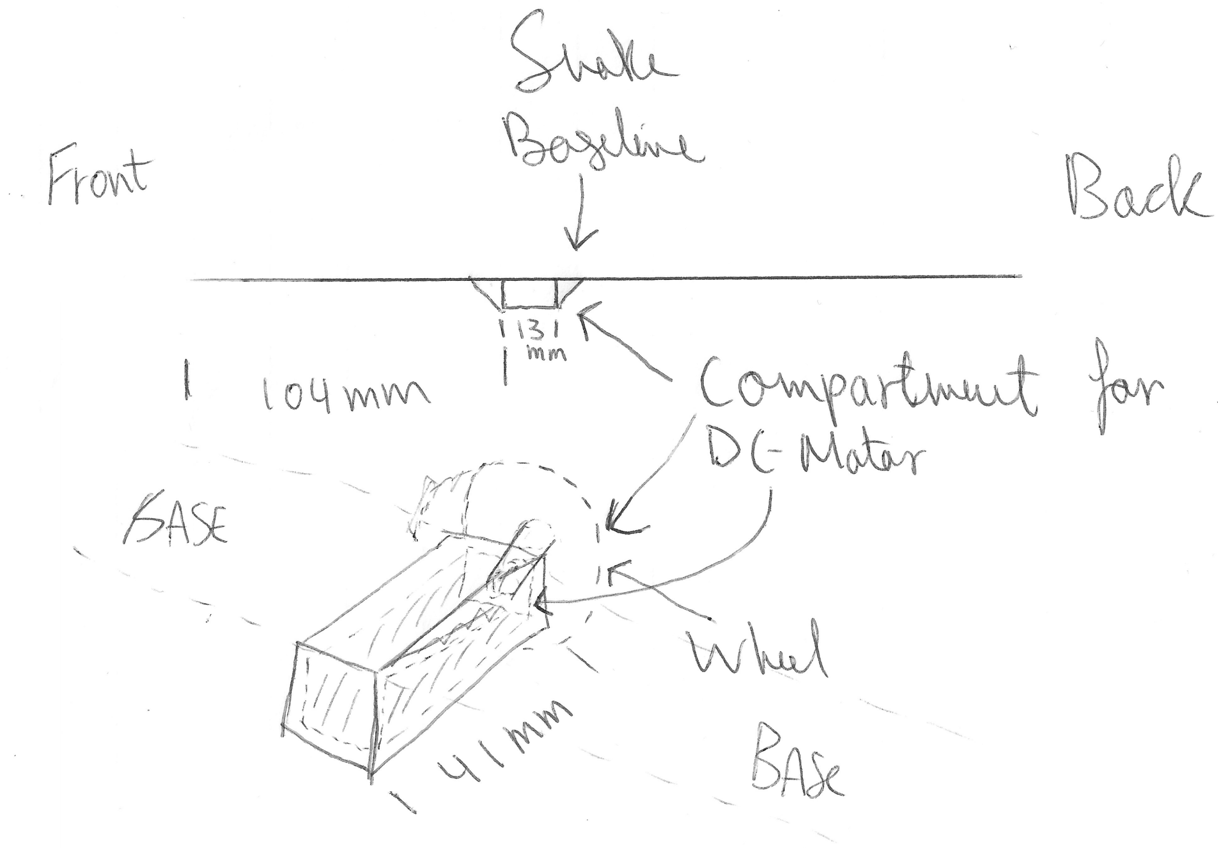


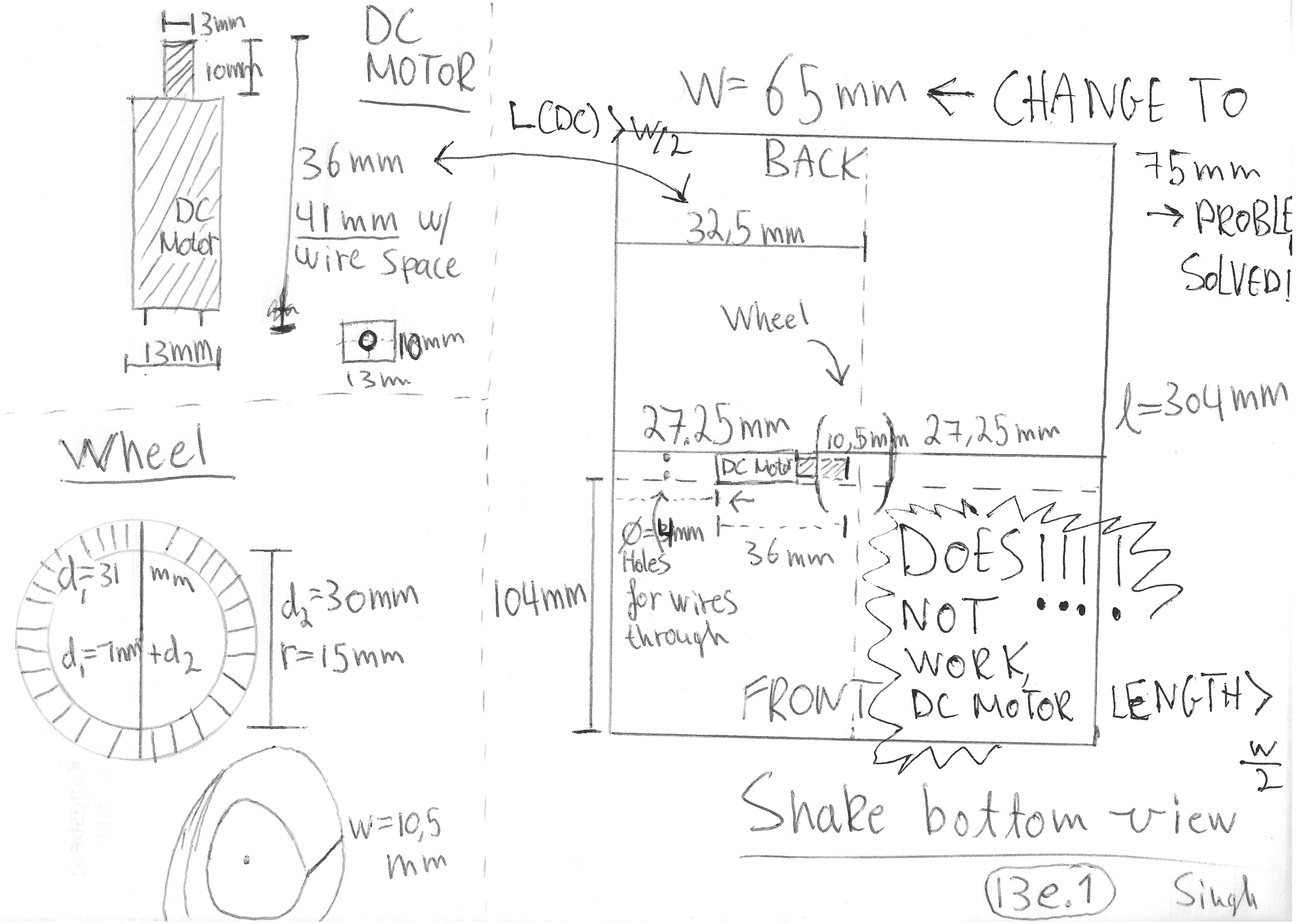
BASE
Whel



15mm
13mm
DC Motor
20mm
15mm
~~18,5mm~~ ~~16,5mm~~
3,5mm
1,5mm
Whel
Compartment

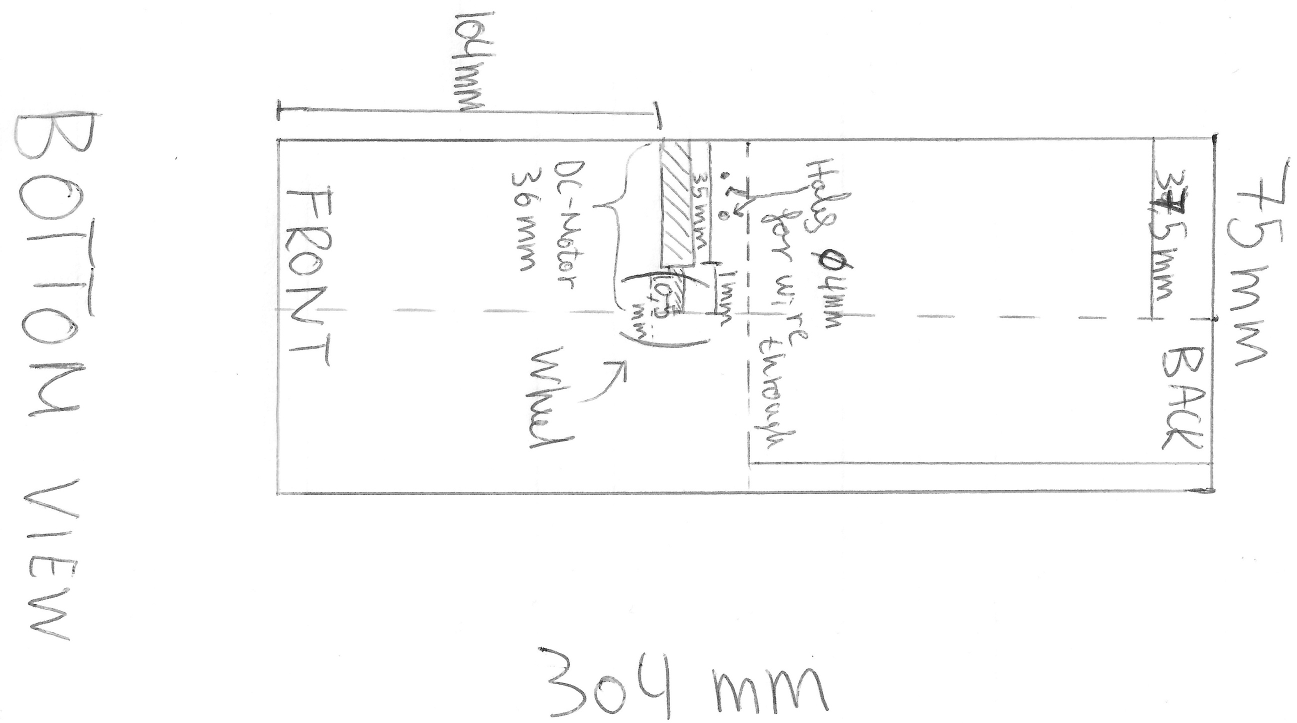
Singh 13d.1


Snake
Baseline
Front
Back
13
mm
104mm
Compartment for
DC-Motor
BASE
Wheel
41mm
BASE



DC MOTOR
3mm
10mm
DC Motor
13mm
36mm
41mm w/ wire space
10mm
13mm
Wheel
d1=31 mm
d1=7mm +d2
d2=30mm
r=15mm
w=10,5 mm
W=65mm ← CHANGE TO 75mm → PROBLEM SOLVED!
L(DC) > W/2
BACK
32,5 mm
Wheel
l=304mm
27,25mm
10,5mm
27,25mm
DC Motor
Ø=4mm Holes for wires through
36mm
104mm
DOES!!! NOT WORK, DC MOTOR ...!
FRONT
LENGTH > W/2
Shake bottom view
13e.1
Singh

If we change Snake width to 75, problem Solved!



13e.2 Singl

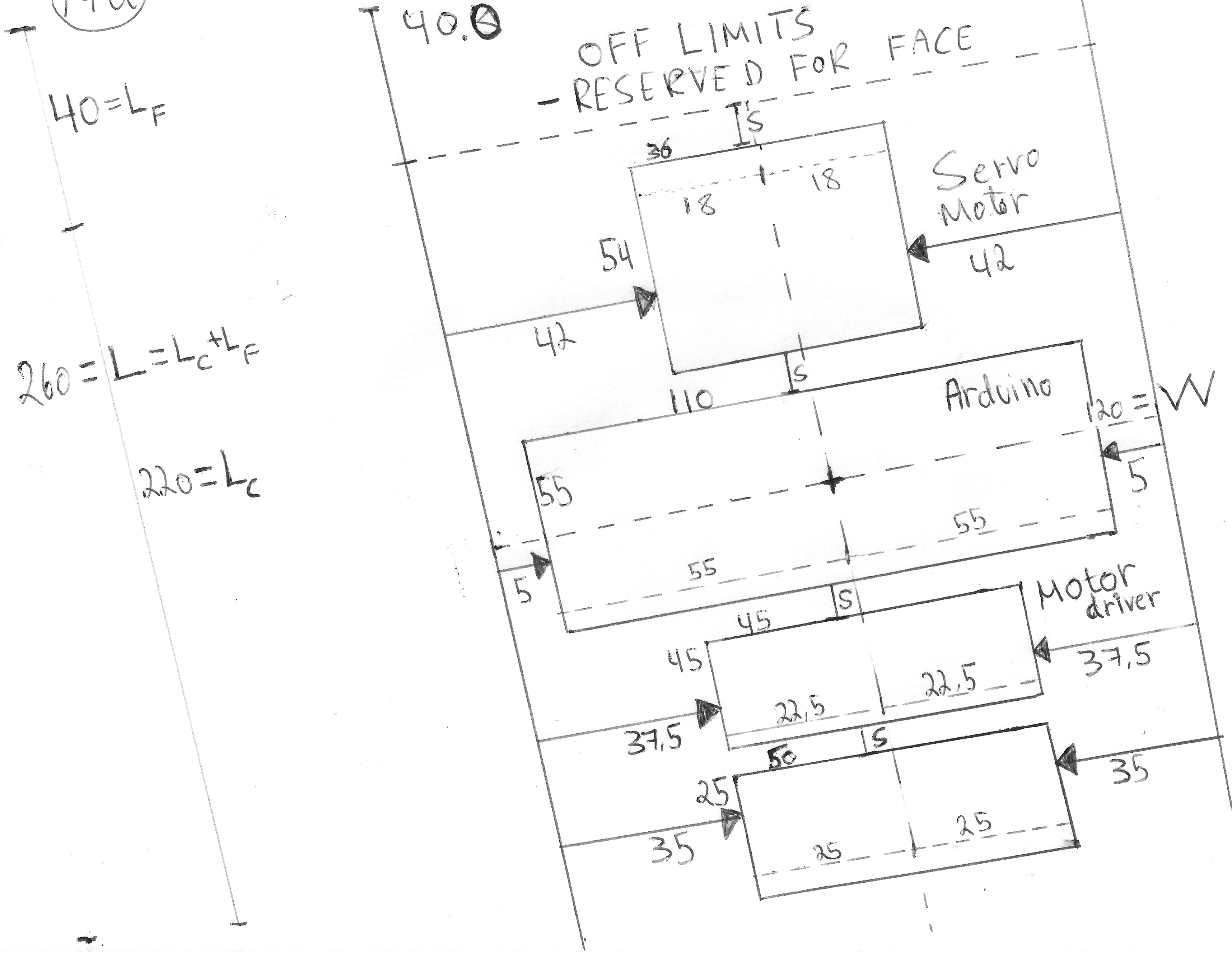
18.03.2020
Singh

ALL UNITS in mm

Placement of Components in Nagini's Head

14a

I_s = 10 Space Clearance

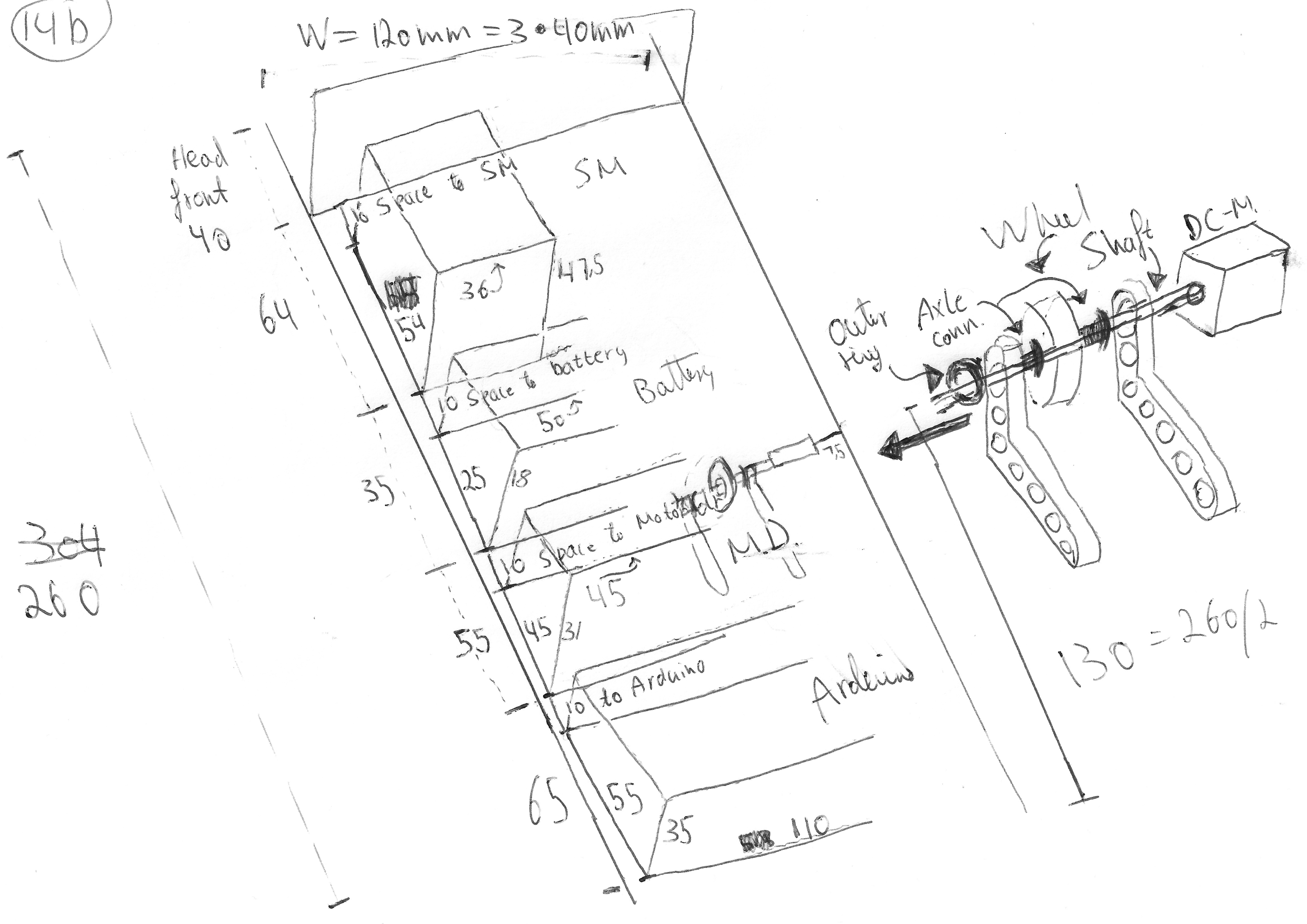


18.03.2020
Singh

ALL UNITS [mm]

(14b)

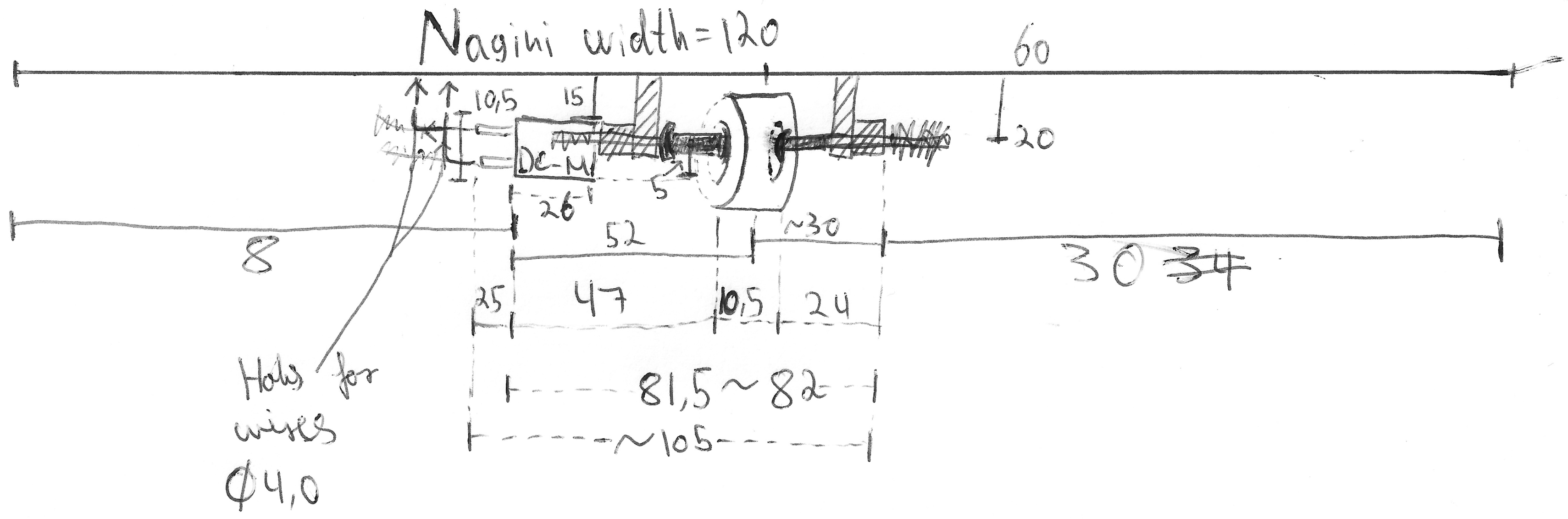
Nagini's head w/movement subsystem



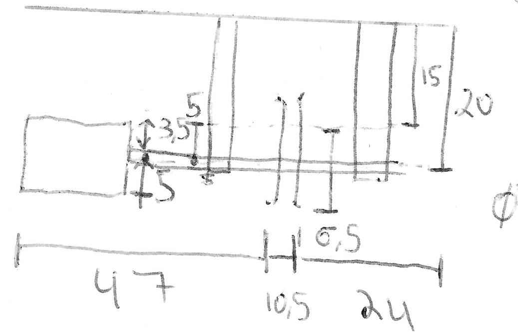
19.08.2020
Singh

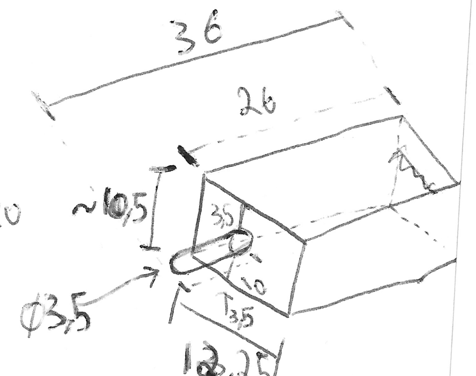
Movement Subsystem placement

15



15a





DRAFT

Singh
And Samuel

Total Components of Nagini V0.1

- 1x Arduino Mega
- 1x 9 or 12V Battery
- 1x DC motors (12VDC, 300rpm)
- 1x Motor driver
- 1x Mini Lidar
- ~~1x Wheels for Snake head~~
- C-brackets (Link) X 4
- Joint X 4
- Snake head (1 link) X 1

5 LINKS

- 5x Servo motors
- 5x Wheels

Appendix B Total equipment

Electronics:

- Arduino Mega 2650
- L298N Motor Driver
- N20 Micro DC-Motor (12V - 300rpm)
- 9V Lipo battery
- VS1838B IR Receiver
- IR Remote
- Seed Mini Lidar/HC-SR04 Sensor
- Hitachi HD44780 compatible LCD w/ I2C backpack
- 4x - TowerPro MG995 Servo Motor
- Half-sized breadboard
- Dupont jumper wires

3D printed parts:

- Links v0.2
- Joints v0.4
- Central Brick v0.3
- Head v0.2
- Face v0.1
- Compartment

Lego modules:

- 6x - Technic Axle 6
- 14x - Technic Beam (90 deg)
- 2x - Long Pin
- 10x - Wheel Hub
- 10x - Narrow Tire with Ridges Inside
- 21x - Half Bushing
- 1x - Axle 3
- 2x - Long Pin with Friction and Bushing

Screws/bolts:

- 12x - M3 (3x10mm) flathead woodscrews
- 8x - M4 (3.8 to 4x25mm) bolt/screw

General equipment:

- Cable sleeve (Ø: 12mm)
- Double-sided tape
- Velcro
- Glue gun
- Screwdriver set
- Drill with accompanying drill bit set
- Mini-hacksaw