Robotics Project 2015 - CDT508
SWARM

Peter Cederblad	pcd11001@student.mdh.se
Albin Barklund	abd11003@student.mdh.se
Roxanne Anderberg	11001@student.mdh.se
Ragnar Moberg	ragnar.morberg@gmail.com
Sudhangathan Bankarusamy	sby14001@student.mdh.se

Märlardalens University
Västerås, Sweden

January 19, 2016

1. Project introduction

This year the project course has four different projects. ROARy, Butler, Swarm and a UAV project. All the students has been divided into pool groups with different specializations. Hardware, which developed all the electronics. Software, they developing and implemented code and simulations for different systems. Communications which worked with all type of communication e.g. how to communicate with the Naiad robot and how to control it. Mechanics they worked with the different structures and CAD programs for all projects. The working hierarchy that has been used are four different project managers and four pool group leaders, with a different amount of workers. The project managers shall treat the pool groups as there was small consult companies and that the project leaders buy a service of them. It is up to the group leader at every pool to distribute the work among his teammembers in order to achieve all the goals that the project managers has set up.
The purpose of this hierarchy is two folded. Firstly, reuse solutions between the projects in order not to invent the wheel all the time. Secondly, this hierarchy has the possibility to be more dynamic than if all students where locked in a small group working on a specific project.

This report will only address the swarm project.
The swarm project consists of two sides. One side consists of the old Naiad platform that was started to be developed in 2013 and the second side is swarm behavior. On the Naiad platform a light communication system for underwater purposes has started to be developed and a Wi-Fi solution for programming the robot has been installed, instead of the old ethernet cable. A second robot has been started to be developed, which is called Naiads sister. The hull and motors are bought, but not assembled. A design for the hull to the new thrusters has been initiated.
The aim for the swarm behaviour side of the project has been on finding an Algorithm for shape forming, the project members has initiated the implementation one.
A new land based platform has been bought called kilobots. The purpose for using a land based platform instead of only the underwater platform, is that the school does not possess a large enough pool on the campus area that can fit more than one robot of the naiad type.

1.1. The structure of the report

The rest of the report has the following structure. Section 2, contains a description of all the goals that has been set up for this project.
Secondly all the milestone for the different subsystems to be developed are located in section 3. Next section, section 4 lists all the specified requirements.
Section 5. The planning section shows the overall plan for this semester in tableform. For a more detailed description of the plan, open the file Swarm_BIS.pod, which is located in the same dropbox map as this report.

Section 6 contains an introduction of the subsystems that has been investigated during this year on the old Naiad platform. As well as an Quick starting guide for the Naiad.
Next section, section 7 describes the new platform that has starting to take form. In this section one can find the documentation of the new IMU card as well as future work on the platform.
Section 8 describes the need for a WiFi solution, what has happen on the wireless communication front this year, a short description on the configuration is also available and lastly there are behavioral-issues and future work.
Section 9 Underwater communication is described, this section is divided into two parts. Firstly there is the a section to del with the electronics that has been implemented. Secondly the communication between two computers using the developed electronic system solution.
Section 10 contains informations information on which students that will work on this during there masters thesis.
The swarm behavior section, section 11 is divided as dollows. A short introduction about what has been done and the purpose for this year. Secondly, it contains a description on the codes that has been implemented for the klobots. Thirdly a description of the Matlab code for the self-organizing coordinate system and the different start up codes in matlab for the next years students. Fourthly a summery of the S-DASH algorithm that still needs to be implemented is provided. and lastly conclusions and future work.
Lastly the Section 12 contains all appendices for this project.
In this report the author's name for every subsection can be found in () after the subtitle. The author is responsible for the text until next name appear. The purpose of this is to make grading easier for the examining teacher. The project leader, Peter Cederblad has putting it all together.

Contents

2. Goals

For the Naiad platform the goals has been to build a second robot. Improve the power supply on the first robot. Go from ethernet cable to a wireless system for communication with the robot in surface mode. Also to develop two different underwater communication systems. One short range but fast with LED technology and one long range but more slow using sonar technology.

3. Milestones

Naiad

- Current system up and running
- Wireless communcation using standard router
- Build new tail antenna
- New power supply
- Underwater light communication up and running
- Side sonar up and running
- Long range underwater communication up and running

Naiad sister

- Body
- Build interial for electronics
- Build all electronics
- Communication interface
- Light Communication between two robots

Wireless communication

- Specification
- Underwater test
- Tail Antenna complete

Light communication system

- Specification
- Land test, Prototype solution

- Underwater test, working solution

Swarm behavior

- Self-coordinate system
- S-DASH in Matlab
- S-DASH on kilobots

4. Requirements

Naiad platform new power supply, ring design.The wireless router shall be small enough to fit inside the current platform. The wireless router must be able to work in the current system without major changes to the power plant. The antenna for the wireless communication system shall resemble a stingray tail. The antenna shall be stiff. The antenna's length shall be 30 procent of the length of the Naiad platform. The toolplate shall be able to mount the side sonars developed by DeepVisionAB in Lindköping. The toolplate shall be build of the same material used on the current platform. The underwater light communication shall be able to work in pool water with a range of 100 m. The long range underwater communication system

5. Planning

The plan for the the Naiad platform this fall has been

- Current system knowledge
- Current system up and running
- Test: water leakproof
- Test: Old system dive

The plan for development of the wireless communication

- Specification for wireless communication
- Develop wireless communication
- Test: Wireless communication system
- Verification and validation on the improved Naiad
- Build tail antenna

The plan for side sonar overlapping with stereovision

- Specification of sonar element for overlapping image with stereo vision

- Order hardware for sonar element
- Design
- Build sonar holder for Naiad
- Implemention: Software overlapping between sonar and Gimme 2
- Test: Sonar and Camera in a big pool
- Verification and validation

The plan for Naiads sister

- Specification on Naiads sister
- Design
- Buy hull and motors
- Build interial for the electronics
- Build the electronics
- Implement new software
- Integrate subsystems
- Test: Water leakage proof
- Test: Naiads sister dive test

The plan for the light communication system

- Specification light communication
- Design light communication
- Develop light communication system
- Test: Blink light communication
- Verification and validation

The plan for the long range communication system

- Specification for sonar communication
- Design sonar communication system
- Develop sonar communication
- Test: Sonar communication

- Verification and validation

The plan for the landbased platform

- Read state of the art
- Specification on the landbased robot platform
- Search for landbased robot platforms
- Order landbased robot platform
- Hands on sytem knowledge on the purchased system

The plan for the swarm behavior

- Read state of the art
- Design: Information repressentation
- Design: Path Planning
- Develop algoriths for swar behavior
- Development of simulator
- Implementation on landbased platforms(Kilobots)
- Test on AUV scenarios
- Verification and validation for AUV scenario 1
- Verification and validation for AUV scenario 2
- Test on landbased platforms()
- Verification and validation on landbased platform

The plan for presentation

- Result of landbased swarm system
- simulation result for AUV scenario 1 and 2
- Result on the light communication system

Chrismas holiday Report writing and time for completion

6. Naiad old system

6.1. Introduction (Peter Cederblad)

For this year a light communication system has started to take place and a wi-fi solution has been implemented. There has also been work done in making a new antenna for the wi-fi connection. We where also given the task at looking into how to integrate a side-sonar on the Naiad that in order to overlap the sonar picture with the stereo vision that is already onboard the platform. Not much work has taken place here, a few students went down to Lindköping and meet the company that are developing this kind of solutions. The companys name is Deep Vision AB. There has also been work in manufacturing a new tool plate in order to be able to use the side-scan system.
All systems that has been developed for the Naiad is also for the sister robot and will not be written about here. Instead all systems has there own section in this report. The mechanics group has written one report that covers all there work during this fall, it can be found in the appendix C

6.2. Software Guide to Quick Starting NAIAD - Sudhangathan Bankarusamy(This subsection)

The NAIAD is mainly controlled by a Beagle Bone Black(BBB) running Ubuntu 12.04. Given below is a series of steps that will start the thrusters with a mission. Before starting with the steps below, care should be taken so that NAIAD is powered-up using 22.5V power source, both kill and mission switch is plugged in its respective slots, the UL(user led) LEDs on all CAN cards must be flashing. Also a computer, preferably a laptop should be connected to the network switch, using a LAN cable or connected to the wifi access point as explained in later sections, so that a network connection from the computer to the BBB exists. The IP address on the computer should be manually set to 192.168.1.10(192.168.1.xxx).

1. From the the computer using a terminal type:
   ssh ubuntu@192.168.1.1
   for password enter "ubuntu"

2. type:
   cd BBB/

3. type:
   sudo system.sh run
   in this step the all the programs necessary for the mission are running

4. In order for the thrusters to start, pull out the mission switch.
   At this point the thrusters should start working

5. Pulling out the kill switch will stop the mission and the mission will be restarted once the kill switch and mission switch are put back.

6. type:
   sudo system.sh kill
   This will stop all programs

6.3. Conclusions (Peter Cederblad)

There are a few systems that has begun to be implemented. However it is obvious that when a project like Naiad, that has a few years on the neck the students are starting to lose their interest in it. Especially when there are a lot of other projects around them that are newer.

6.4. Future work

On the Naiad platform there are several things that needs to be improved.

- Light communication system
- Sonar communication system
- Side-scan with stereo vision
- Wi-Fi antenna
- Toolplate

These are all subsystems that still has lots of work.

7. Naiad sister

7.1. Introduction

The second Naiad has begun to take place. At least the major part of the hull and the thrusters has arrived to the school. Unlockely there has not been so much time in order to assemble the hull or the thruster hulls. A new power supply has been discussed, but the students in charge of that has not documented it. The same for the thruster hulls. But an new IMU card has been made to the new IMU.

7.2. IMU card - Roxanne Anderberg

7.2.1. Description and Requirements

For measuring yaw, pitch and roll, Naiad uses an inertial measurement unit. In this case VN-100 Rugged [2] from Vectornav [1]. VN-100 Rugged can communicate through TTL or RS-232 and since Naiad's CAN card has UART the decision of making a UART RS-232 adapter was made. Luckily there are transceivers on the market for just that kind of adaption.

7.2.2. Design and Interface

For making an UART RS-232 adapter you only need a transceiver [4] and the matching capacitors for that transceiver. On this card I have also made pinouts that have been carefully measured so the card can match the CAN cards perfectly as an extension card. The second and final version of the IMU card have shorter sides so that the communication group can reach the pins for the debugging on the CAN card without having to remove the IMU card. There also been added test points for testing the transceiver.

7.2.3. Testing

For testing the transceiver to make sure it is not broken I used a few steps found on a forum [5]: "Troubleshooting this IC is not difficult. Power up the MAX232 and check voltage across supply pins to insure correct voltage input. The following two tests check each "TTL in - RS232 out" converter. Put 0 volts into pins 10 and 11, then check pins 7 and 14. Should have about 10 volts output on each. Put 5 volts into pins 10 and 11, then check pins 7 and 14 again. Should have about -10 volts output on each. If that succeeded, then check each "RS232 in - TTL out" converter. Connect pin 7 to pin8. Connect pin 13 to pin 14. This will be a "loopback test". Put 0 volts into pins 10 and 11, then check pins 9 and 12. Should have about .6 volts output. Put 5 volts into pins 10 and 11, then check pins 9 and 12 again. Should have about 4.5 volts output."

For testing the circuit you can use an oscilloscope on the UART pins and the RS-232 pins to make sure that they receive/send the right data.

7.2.4. Using the system

The card is very much plug and play, but before using the IMU it is very importing to read the manual. There suppose to be manuals in the school but one can also find them online [3].

7.2.5. Future Work for IMU card

The card that has been printed is not a card that should be used because it has some faults. Find a good MAX232, there is many versions. Test it on a breadboard first to see if the direction of the capacitors are the same as the datasheet. For printing out in school, make sure to have the traces to the hole mounted components on the other side of the card. Will be a bit hard to solder otherwise. Also future work is the cabeling between the IMU and the PCB. The Harwin connector, see table 4, is very expensive and difficult to attach. The IMU must have the female version but for the card I would recommend another type of connector.

7.3. Conclusions(Peter Cederblad)

There are a lot here that can be improved from the first robot. No need to rush it and make bad decisions.

7.4. Future work Naiad sister

Everything.

8. Wireless Communication

8.1. The need for the WiFi module(Sudhangathan Bankarusamy)

The cables used for the NAIAD AUV had limiting issues such as the range, drag due to the cable running through water, which also adds up energy expenditure. Moreover the cables were an alternative to the limiting under water wireless technology issues. The Wi-Fi router is a starting point for trying various wireless technologies.

8.2. Network configuration

Configuration without wireless router was a simple star network where a gigabit switch is used and all devices can connect to it using a LAN cable. In the configuration with the Wi-Fi device(an ASUS WL-330n) is used in 'access-point' mode. This access point is used as a gateway between wired and wireless devices. The DHCP range in it's configuration page is set to 192.168.1.3 to 192.168.1.99.
The IP address of BBB remains as 192.168.1.1 and that of the access-point is set to 192.168.1.2.

8.3. Behaviour and the issue

The router performed well as expected, but only as long as the NAIAD AUV is not under the water surface. As the Wi-Fi signal deteriorates heavily through water, the connection is immediately dropped when the AUV is submerged in water.

8.4. Future work

Implement OpenVLC - OpenVLC is an embedded framework for visible light communication. It has the complete software stack, from MAC to application layer libraries. OpenVLC can run in Linux on BeagleBoard. After setting up OpenVLC, an interface VLC0 is created through which we can send/receive ethernet data. The physical media is light through the air. The prototype setup is further discussed in Underwater Light Communication section.
SONAR - Sonar is the most used technology for under water communication. It has been shown to perform well for long distance under water communication.

9. Underwater communication

9.1. Electronic part (Albin Barklund)

9.1.1. Description

The light communication module is a wireless data link for short distance underwater communication between autonomous underwater vehicles (AUVs). It consists of a receiver and transmitter which utilizes power LEDs and photoresistors to send and receive data.

Bear in mind that no actual solution have been implemented and this document only summaries a couple of experiments which where conducted to prove or disprove the hypothesis in section 9.1.3.

9.1.2. Requirements

Range:	2 - 3m
Data rate:	1 kbit/s
Input data signal:	5V
Medium:	water

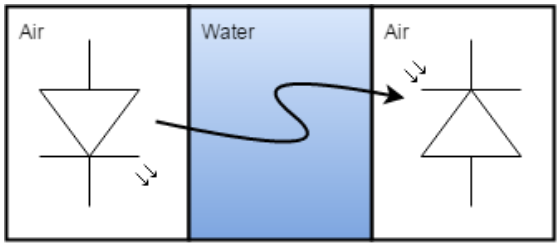


Figure 1: Signal transistion through mediums

9.1.3. Hypothesis

With the use of power LEDs and photodiodes it is possible to create a wireless data link for underwater communication that modulates bits simply letting an asynchronous data signal control a LED which fulfill the requirements in section 9.1.2

9.1.4. Design and interface

Transmitter
The transmitter takes an asynchronous data signal as input. The driver stage then inverts and amplifies the signal which then drives the LED. If the signal is high then the LED is turned off and if the signal is low then the LED is turned on.

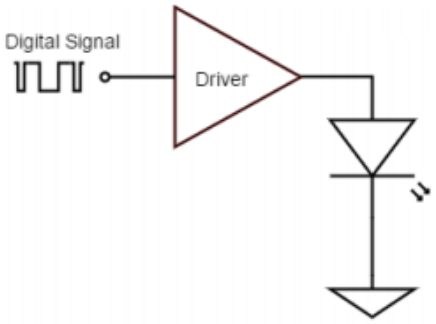


Figure 2: The transmitter

Receiver

The flashing pulses from the LED excites the photodiode and converts the light into current. The current is then converted to a voltage and passed through an amplifier. Since the signal to noise ratio is very low both ambient noise and fluorescent light needs to be filtered out from the signal which is accomplished by the high pass filter. Finally the signal gets reconstructed through a schmitt trigger.

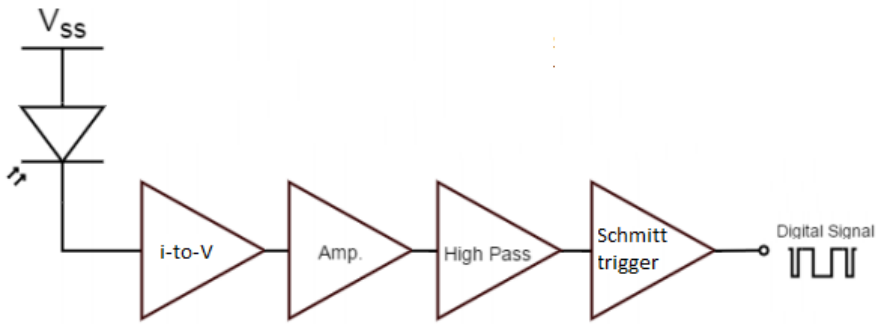


Figure 3: The receiver

9.1.5. Test and simulation results

The system has proven capable of transmitting asynchronous data with a baud rate of 4800 within a range of 2 meters on land. If a rectangular cointaner filled with water were put in between the transmitter and receiver the range decreasesed to 1 meter.

The transmission only succeeded if data were sent continuously. If a short break would occur in the continues data stream it takes approximately 20 bytes before the receiver starts outputing valid data again.

Figure 4: Filter paramter variation 1 and reconstructed signal

Figure 5: Filter paramter variation 2 and reconstructed signal

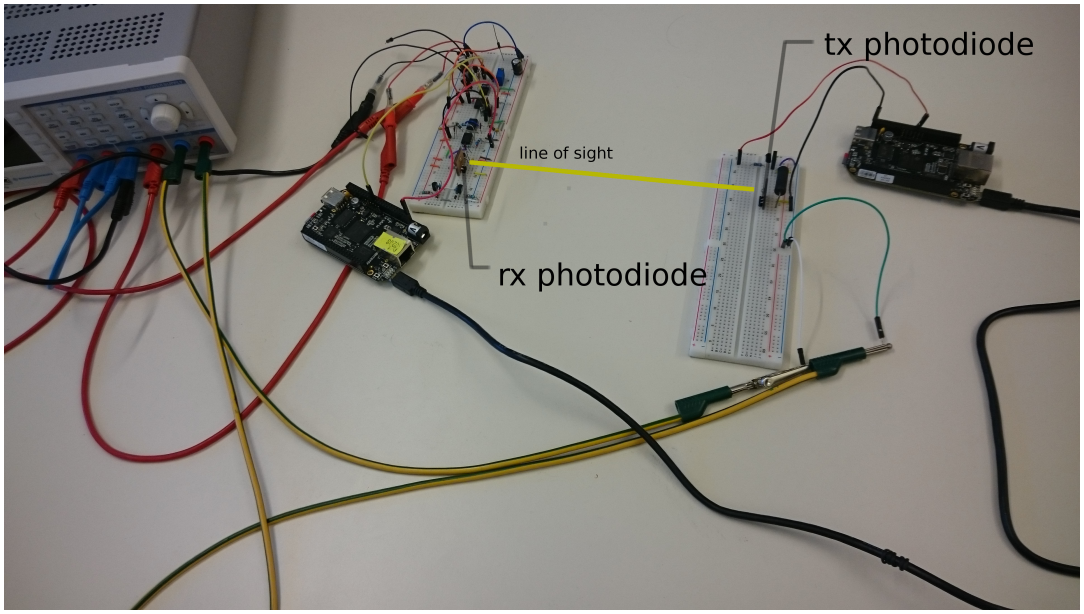


Figure 6: Prototype: Picture of the lab setup

9.2. Communication part (Sudhangathan Bankarusamy)

9.2.1. The prototype

This setup consists of an LED and photodiode, each of them connected to a Beagle Bone Black(BBB). The LED and photodiode electronic circuits are explained in the electronics section. Figure 6 shows the actual setup. In this prototype version the OpenVLC framework is not used for the lack of time. Instead a simplex serial communication is emulated using light over air.

9.2.2. Transmission and receiving of information

The bits of information is sent to the LED over the serial port, TX, of the BBB. While on the receiving side a photodiode is connected to the RX of the BBB. Any application that uses serial port to transmit information can be used with such a setup. In this case, minicom was used to send and receive characters. Both the BBBs were connected to a laptop. Two different serial terminals were used to connect to both the BBBs and the character send/receive experiment is done.
Table 1 shows the setup parameters.

9.3. Conclusion(Albin Barklund)

The method works and the technology could probably be used for under water communication even tough it is most likely not a good solution. This is due to the fact arbitrary

Table 1: Prototype parameters

Parameter	Value
Baud rate achieved(stable)	2400 bps
Distance (stable)	1.6 m
LED power	1 W
Wave length	473nm(Blue)
Rx pin number(RX BBB)	P9:23 (UART1_RXD)
Tx pin number(TX BBB)	P9:24 (UART1_TXD)

digital signals contain many different frequencies and the modulation scheme suggests sending all of them through the medium.

9.4. Future work

A better approach would be to implement a data link with LEDs and photodiodes which uses a FSK modulation scheme instead.

10. Sonar communication(Peter Cederblad)

10.1. Introduction

In order for the underwater robots to be able to communicate for long distances, a sonar communication system needs to be implemented. Only a basic prestudie has taken place in this subject during this fall. It has been decided that this shall be Albin Barklund and Daniel Adolfssons master thesis work.

11. Swarm behavior

11.1. Intoduction (Peter Cederblad)

In this part of the project there has been an investigation in collaboration between robots. Especially how to form shapes. This will be useful for the e.g. the schools underwater robots that will search and mark oil leakage.
The first thing any robot collective needs is a coordinate system, that has been the first of two things that this year has focused on. The second thing is to make a good starting point for the next year in terms of producing startup code for the kilobots, in c and to create a simulation environment in Matlab

11.2. Kilobot code

For a land based platform the kilobot platform [8] was chosen after a few weeks of investigation on different platforms. For the kilobots there has been produced a number

of small programs for the students to start with, in order for them to get a quick introduction on how the kilobot API works [8]. These small programs consist of finished exercises. Exercises that was written for a different library [7], than the one that followed with the kilobots when they were purchased.
A short description will be included her for each program.

- **Lab0_Blinky.c**
  This is the first program to start with. It introduces a basic function and the purpose of this program is to show how to manipulate the onboard LED. Functions used are: Robot_ID(), set_color(), _delay_ms() and also in the main function init_robot() and main_program_loop().
  The main function is the same in all programs and should never be changed, except with which function name the main_program_loop() should call.

- **Lab1_1_Simple_movment.c**
  In this program the motors are added. They use two different function.
  spinup_motors() and set_motor(). Here the robot will move forward for a while and then turn in both directions. The robot will also turn on the LED in different colors.

- **Lab1_2_Nonblocking_movment.c**
  The non-blocking program does the same thing as the first program, except this one take uses interrupt routines.

- **Lab2_1_Test_speaker.c**
  Here the communication begins. The function message_out() is introduced now and the variable enable_tx. Which tells the robot that it is allowed to transmit messages. The robot will broadcast a fixed message. Program one robot with this program and another as a listener.

- **Lab2_2_Test_listener.c**
  This is the second part in the communication. The receiving robot. The get_message() function is used to collect messages and put it in the message_rx array. The program will also blink when a message is received.

- **LAB2_3_Test_speaker_mod.c**
  Now timeing is added. This program uses the variable kilo_ticks which can be located in the libkilobot.c file, inorder to keep track off the time. It will then transmit a 0 or a 1 every other time. And off course blink in different colors depending if it is a 0 or a 1 in the message.

- **Lab2_4_Test_listener_mod.c**
  The message array is further explored in this program. First the robot looks if the number is even or odd, it the reads the distance from the sending robot. The LED is turn on and off in different colors depending on those to factors.

- **Lab3_Putting_it_together.c**
  A multitasking robot is created here. It both sends and reads messages. If the robot receives a message it will start its motors in a random direction. This will make all robots, with time, lose connection to all there friends. When the robot does not receive any new messages it will stop and turn the LED white.

- **Lab4_1_Orbit_star.c**
  This exercise requires two robots. One that can reuse the code from Lab2_1_Test_speaker.c, this robot will akt as a star. The other robot will receive messages from the star and it will try to move around the star in a circle with a fixed distance. This program in it self does not provide any real new functionality, but it is a good program to build more of. Add more robots and make them all move around the same star without crashing into one another.

- **Lab5_Move_to_light.c**
  The kilobots also has an ambient light sensor. In this program the robots will try to move towards the light. This requires a dark space. four new function is used here. Mean_Ambient_Light(), kprinti(), kprints() and abs().

- **Lab6_2_Gradient_simple.c**
  This program is the first one that can be called a little more advanced. Here the robots calculate distance in form of information hops from one predefined seed robot. Each robot will indicate the distance by turning on different colors on the LED.

- **Lab6_Gradient_adptive.c**
  This program is similar to the above, but here a timer is implemented as the gradient can change with time.

- **Lab7_sync.c**
  In the sync program all robots start up with a random delay, and then makes a blink in a specific period, they also sends out a message so the neighbors know when the robot blinks. All robots will then change a little bit on the period of blinking in order to try to match up with each other. With time they will start to blink in sync.

There has also been created a .c file whith some start up code and the structure for the S-DASH. It is called SelfAssemblySelfHealing.c
Functions that has been added to the predefined libkilobot.c are the following.
The four first are more general for the kilobots, while the purpose of the rest is for the S-DASH algorithm:

- **extern int RobotID(int modulusWith)**
  This function collects sensor data from the ambient light sensor. It will then use this data to create a random number, which will be modulated with the input integer value. The function returns an integer value.

- **extern void spinup_motors(int num)**
  For the kilobots to be able to move, they have to overcome the static friction between the legs and the surface. This function does just that. It will set the motor/motors on maximum value for 10ms. After this function the SetMotion() function shall be called. The input is an integer corresponding to which motor to turn on or both.
  - 1: Both motors, this will make the robot ready to move the forward direction.
  - 2: Right motor, this will make the robot ready to rotate in a counterclockwise direction.
  - 3: Left motor, this will make the robot ready to rotate in a clockwise direction.

- **extern void SetMotion(int current_motion, int Direction)**
  This is the second function that has to be called in order to make the kilobots to move. It works in a similar way as spinup_motors(), but the motor values are lower. The four global variables cw_in_straight, ccw_in_straight, cw_in_place and ccw_in_place are used. These variables can be tuned in order to make each kilobot move the desired bahavior.

- **extern int Mean_Ambient_Light(int num)**
  This function takes in an integer value. This integer is the amount of samples that shall be collected from the ambient light sensor. The function returns the mean value of all samples.

- **extern int fakultet(int N)**
  Calculates the factorial of the input integer and returns that number.

- **extern void SwapChar(char* str,int i, int j)**
  Change place on two characters in one array

- **extern void SwapInt(int arr[],int i, int j)**
  Change place on two integers in one array.

- **extern void PermuteStr(char* Ans, char* string, int start, int end)**
  The purpose of this function is to calculate all combinations of neighbors when checking which neighbors that can for a referensgrupp for the triangulation as well as for merging groups in the S-DASH algorithm.
  It takes in an string and prints out all combinations on the computer screen using the kprints() function.
  This function can be changed inorder to store the values for the robot to use instead of printing them on the screen

- **extern void PermuteInt(int Ans, int array[], int start, int end)**
  This function does the same thing as the above, but with and array of integers.The reason for the two implementations, is because the students for the up coming year can choose how to represent the neighbors.

- **extern void CheckAllNeighborsID(int neighbor_ID[], int neighbor_Dist[])**
  This function takes in an robot id value and the distance fromthat robot, it then

All the above functions can be found at the bottom inside the libkilobot.c file.

11.3. Matlab code

In this project Matlab 2015b has been used. There has been 3 major .m programs produced in this project.

- *NetWorkControlSystems_test.m*:This file shows how to solve some basic problems in swarm robotics; Solving the rendezvous problem is one of them. Also how to form basic constellations and make that constellation change direction, move around in a user input way and also follow a predefined path. This file is mostly for getting into swarm thinking, using ordinary vector programming in Matlab. Here only Matlab's own coordinate system is used.

- *ClassTest.m*: Here we take the next step and start looking at objectoriented programming with Matlab. This program solves a little more comlicated problems. First all agents are wandering around randomly untill they see some one else. When this happens they calculate the rendezvous point between all agents that see each other at that area. If there are several small groups created in the Matlab world, they do not take each other into account in this calculation. When a group of agents are close enough to each other they will start to wander in some predefined direction. Here there are room for future work. They should continue to search the environment for more agents. Three different end conditions could be implemented.
  - The maximum number of agents are known and the seach continues untill all all agents belong in one consistens group.
  - A search time counts down and resets for every new agents that are found, when search timer reaches zero. All search groups defines the world population as there own group. They can after this start to solve what ever task the human has given to them.
  - Same as number two, but with an extra deadline that can not be reset. This would mean that the maximum time that the agents has can not be broken, this resembles a search and rescue setup.

- *SelfOrganizingCoordinateSystemTest*3_6.*m*: This is the most advanced .m file and it is crucial for the swarm behavior part of this project that it is finished. The purpose with this program is to construct a consistent coordinate system between all agents that can communicate with each other and measure distances. This is not fully implemented in a robust way. It is possible to get a consistent coordinate system in some special cases, but more work has to be done here.
  The purpose to have a local coordinate system that all agents can agree upon, is that it will be easier to solve tasks together if every one in the group knows were they are all the time, without the use of external coordinate systems as GPS.
  This self-organizing coordinate system is the first part of the S-DASH algorithm.

11.4. S-DASH

This will be a short deskripten of the scalable - distributed self-assembly and self-healing algorithm (S-DASH), for more detaljs see [6].
The S-DASH consists of four main parts.

- First it develops a **consistent coordinate system** that all agents can agree up on. First a set of at least two seed agents must be chosen and then these seed agents will choose two more reference robots. The second part is to use trilateration, which in geometry is a way to determine relative and absolute points when the agents can measure distances. Now there will be at least two local coordinate systems, one for every seed agent. Next step is to merge all local coordinate systems and create a transitional coordinate system that all agents in the collective can relate to. This is implemented in the $SelfOrganizingCoordinateSystemTest3\_6.m$ file and all steps in this test file is well commented. How ever the simulation file needs to be reviewed, it does not have a robust behavior in the end for agents that do not sit in Matlab's orion.

- The second part is **DASH**, the inputs for this function are the desired shape of the collective, given as a picture or a pixel map(black and white, white pixels belong to the shape), desired scale of the shape, robot location in the self-organized coordinate system and messages from neighbors. It will then use the gradient map to decide how to move. It chooses what to do after calculating:

$$\theta_{gradient} = atan\frac{A}{B} \tag{1}$$

  Where gm is the gradient map and
  $A = gm(x_{index}, y_{index} + 1) - gm(x_{index}, y_{index} - 1)$
  $B = gm(x_{index} + 1, y_{index}) - gm(x_{index} - 1, y_{index})$

  Then it has five modes: Gradient follow, trapped robot message, trapped robot moment, random moment and stop movement. Each pixel in the pixel map can be indexed by two numbers(x, y), where y represent how many pixels there are above current pixel and x represent how many pixels that are to the left of this pixel. This will give the upper left corner$(x, y) = (0, 0)$. There are two constraints here.
  - connected shape.
  - All pixels on the outside border must be black.

  Next we have the scale, S_f, defines in terms of robot radius, R_{robot}. IF the shape is a square $3x3$ pixels. AND $S_f = 2.0$ THEN each pixel will be $2 * R_{robot}$ wide and $2 * R_{robot}$long, this means that the dimensions of the real shape will be $6 * R_{robot}$ wide and $6 * R_{robot}$ long.

- The third part is to determine the **scale** of the scape that is proportional to the number of agents currently in the collective.

Every time step every robot updates it's $S_f$to the mean of it's own and all neighbors. After this the robot communicates this to all it's neighbors. IF the scale is too big then all robots has to reduce there scale. This is done in three steps.

- A distributed mechanism to let all robots know that the seed position is unoccupied.
- A way to know how long time to wait until the seed position should have been taken.
- A mechanism to reduce the scale.

Detecting unoccupied seed: $T_{unoccupied}$ updates as following. $T_{unoccupied}$ is compared with all neighbors $T_{unoccupied}$. IF there are any neighbors value that is lower then its own value of $T_{unoccupied}$ THEN $T_{unoccupied} = neighborsvalue + 1$. IF no neighbors value are lower THEN increase own value with 1. IF the robot is located in the seed position THEN set $T_{unoccupied}$ value to zero and update all neighbors.

Above pseudoAlgo will have the effect as a timer. If no robot is located at the seed position, then all robots will start to increase there $T_{unoccupied}$ value by one for every loop in the main controller. Eventually one robot will reach $T_{unoccupied_{max}}$ which will tell the robot that the scale is to big.

The $T_{unoccupied_{max}}$ is calculated as followed.

$$T_{unoccupied_{max}} = \frac{S_f * L_{externalpath}}{V_{robot} * P_{move}} \tag{2}$$

This means: The worst case of traveling divided with the average moment speed.
where,
$S_f = Currentscalefactor$
$L_{external_path} =$
Max(The distance between the starting pixel outside the shape and all pixels on the border of the shape)
V_{robot} = Speed of the robot movement
P_{move} = The probability that the robot will move

Reducing the scale: The scale reduction is initiated by the first robot to reach $T_{unoccupied_{max}}$ or go above it. NOW this robot will do two things:

- Set it's own $T_{unoccupied}$ value to 0
- For $\frac{T_{unoccupied_{max}}}{2}$ cycles of the main control loop, do not use the ordinary scale update function, instead report to all neighbors the new lower value $S_{f_{new}}$. This will effect all robots S_f value.

  The $S_{f_{new}}$ is calculated as following:

  $$S_{f_{new}} = S_{f_{old}} * sqrt(1 - (\frac{1}{NumPix})) \tag{3}$$

  where NumPix is the total number of pixels in the shape pixel map's shape segment.

However after that the scale has been observed for a long time and it is not big enough, three things needs to take place, similar as when the scale is to big.

- A distributed mechanism to let all robots know that the external segment seed position is occupied.
- A way to now how long time to wait until the seed position should have be free.
- A mechanism to increase the scale a proper amount.

Detecting occupied seed: $T_{Occupied}$ this variable is updated every loop in the main controller, just as $T_{unoccupied}$. In every cycle the robot checks to see if it is in the external segments seed position. IF true, then $T_{Occupied} = T_{Occupied} + 1$. IF the robot receives a message called "$moved\_into\_shape\_message$" then it will put a zero in the $T_{Occupied}$ variable. IF $T_{Occupied} < T_{Occupied_{max}}$ then it is time to increase the scale. The robot will also send out the $moved\_into\_shape\_message$.

Wait time: The definition for $T_{Occupied_{max}}$ is the same as the definition of $T_{unoccupied_{max}}$ except for the $L_{internalpath}$, which is defined as: Max(The distance between the starting pixel inside the shape and all pixels on the border of the shape)

Increasing the scale: The robot that initiate the increase scale do two things:

- Send out the $moved\_into\_shape\_message$, to prevent other robots from further trying to increase the scale by resetting the $T_{Occupied}$ values.
- For $\frac{T_{Occupied}}{2}$ cycles of their main controller loop, they do not use the original update function(same as above) instead they send out a new larger scale value $S_{f_{new}}$. $S_{f_{new}} = sqrt(S^2_{f_{old}} + (Pi/NumPix))$.

- The last and fourth part is to **choose a role** based on its location within the shape and with respect to time(partial-temporal differention) This is a distributed method to allow each robot to choose a role based on its location in the desired shape and time. The overall choice of all robots in the shape is called the partial-temporal role pattern. This pattern is given to each and every robot in the form of a pixel map or an image, called the partial-temporal role map. Inputs for this function are the spartial-temporal role map(or just a Spartal role map), desired scale and the robots location in the coordinate system. The function shall do the folowing: All robots has either an Spartal role map(1) or Spartal-temporal role map(2). IF(1) THEN each entity in the map consists of just an integer value, which represents a role.

  In the above example there are to different roles, "1" and "2". The outer part has role number "1" and the inner robots will have role "2". This could represent different colors on the leds or other things.
  IF(2) THEN each entity in the map consist of a list of paired numbers. A timestamp and a role.

In the above example: At time = 0, The robots in the left halv of the square should

Table 2: The spartial role map

1	1	1	1
1	2	2	1
1	2	2	1
1	1	1	1

Table 3: The spartial-temporal role map

(0,1) (1,1)	(0,1) (1,1)	(0,2) (1,1)	(0,2) (1,1)
(0,1) (1,2)	(0,1) (1,2)	(0,2) (1,2)	(0,2) (1,2)
(0,1) (1,1)	(0,1) (1,1)	(0,2) (1,1)	(0,2) (1,1)
(0,1) (1,2)	(0,1) (1,2)	(0,2) (1,2)	(0,2) (1,2)

choose role "1" and the robots on the right side will take on role "2". This will be the first () in each entity. The second () in each entity corresponds to the second timestep.

11.5. Conclusions

The simulation environment for the self-organizing coordinate system took far more time to develop then expected. Things to note about the experimental design is that the test cases shall not just be to small which is common knowledge, but also the minimum size of the testcases had an hugh impact.

11.6. Future work

Finish the implementation of the S-DASH algorithm, both in Matlab and in c code.

12. Appendices

A. IMU - Bomlist(Roxanne Anderberg)

BOM					
Name	Vendor	Ordernr	RefDes	Value	Shape
Electrolyte Capacitor	Würth	865230640001	C1	0.1uF	CAPAE430X540N
Electrolyte Capacitor	Würth	865230640001	C2	0.1uF	CAPAE430X540N
Electrolyte Capacitor	Würth	865230640001	C3	0.1uF	CAPAE430X540N
Electrolyte Capacitor	Würth	865230640001	C25	0.1uF	CAPAE430X540N
Harwin male connector	Farnell	1144561	IMU_CON	HDR2X5	Harwin_Male
LED SMD	School		LED4	LED_blue	INDC1608X95N
2.54 mm pin header	Würth	61300411121	POWER	HDR1X4	HDR1X4
Resistor SMD	School		R24	2.2kOhm	RESC1608X63N
2.54 mm socket head	Würth	61300211821	TEST_HEADER	HDR1X2	HDR1X2
2.54 mm socket head	Würth	61300211821	TEST_JP	HDR1X2	HDR1X2
Tranceiver	Farnell	9724370	U12	MAX232AEWE+	SOIC127P1032X265-16N
2.54 mm pin header	Würth	61300411121	UART1	HDR1X4	HDR1X4
2.54 mm pin header	Würth	61300411121	UART2	HDR1X4	HDR1X4

Table 4: BOM

B. IMU - Schematics and CAD

The schematics have made in multisim and the CAD files have been made in ultiboard.

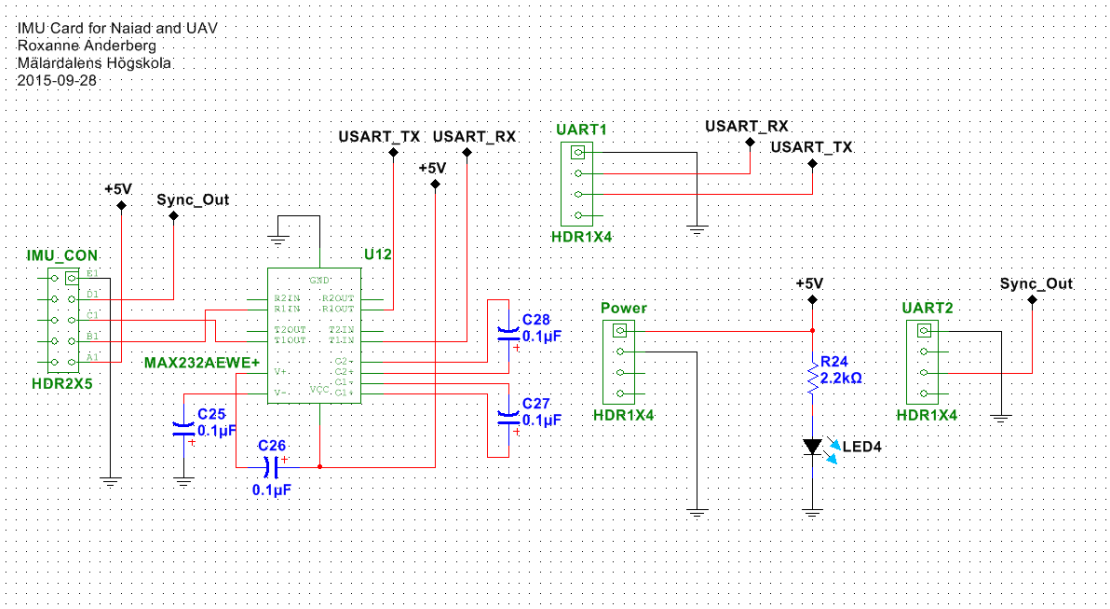


Figure 7: Schematics for the UART RS-232 adapter

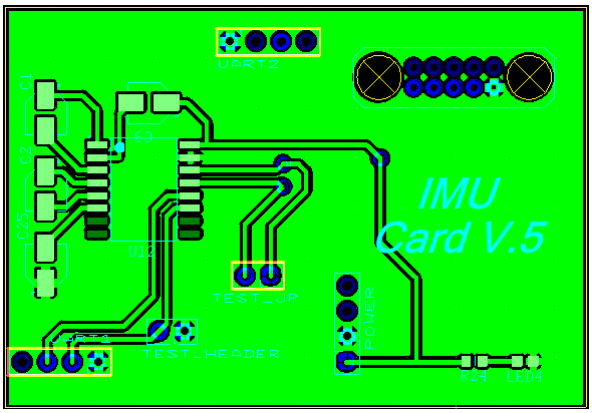


Figure 8: Top Layer

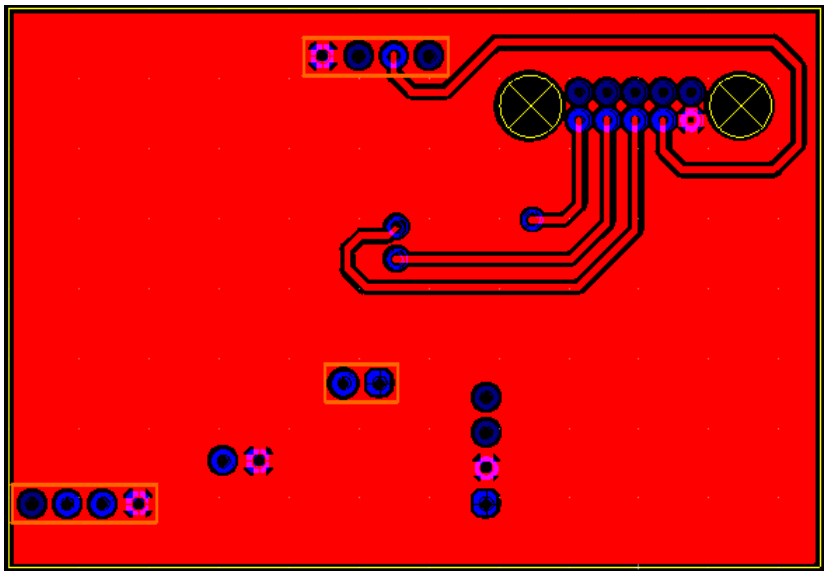


Figure 9: Bottom Layer

Figure 10: CAD model of the extension card

C. Mechanics Group's report(Ragnar Moberg)

Ragnar Moberg igår 14:57

Till: Peter Cederblad

Re: Rapport material till Swarm

RM

As the mechanics group has only put a limited amount of time on the Naiad-project not much work have been done for mechanics on Naiad during this semester, other than that a few missing screw has been replaced. There exists an untouched second pre-manufactured chassis and set of motors with intended use for building a second Naiad.

In order to CNC-mill a tool plate we have found two companies that can possibly sell material and their websites are "teknipur.se" and "obow.se". We have not got confirmation if their material is feasible for underwater use. Since the tool plate has not been manufactured it would be a good idea to redesign it slightly for the acoustics modem before manufacturing.

In order to make a Wi-Fi antenna that point upwards from the back of Naiad we have two ideas. A stiff hose could be bent into desired shape or multiple parts could be printed out in an elastic material and be assembled.

Den 12 januari 2016 15:20 skrev Peter Cederblad <peter_cederblad@hotmail.com>:

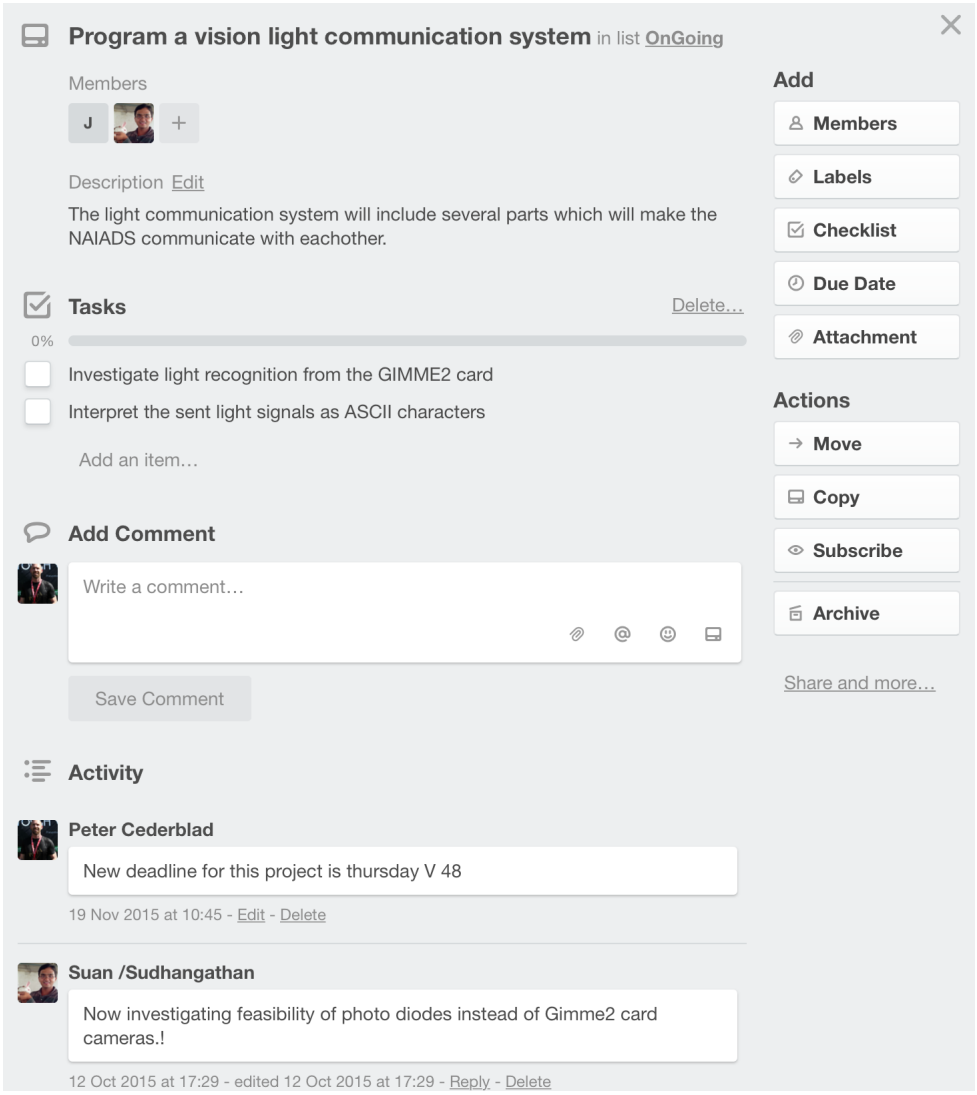
> Skicka över ert material för Swarm rapporten senast Torsdag kl 1500, så jag kan sätta ihop allt.

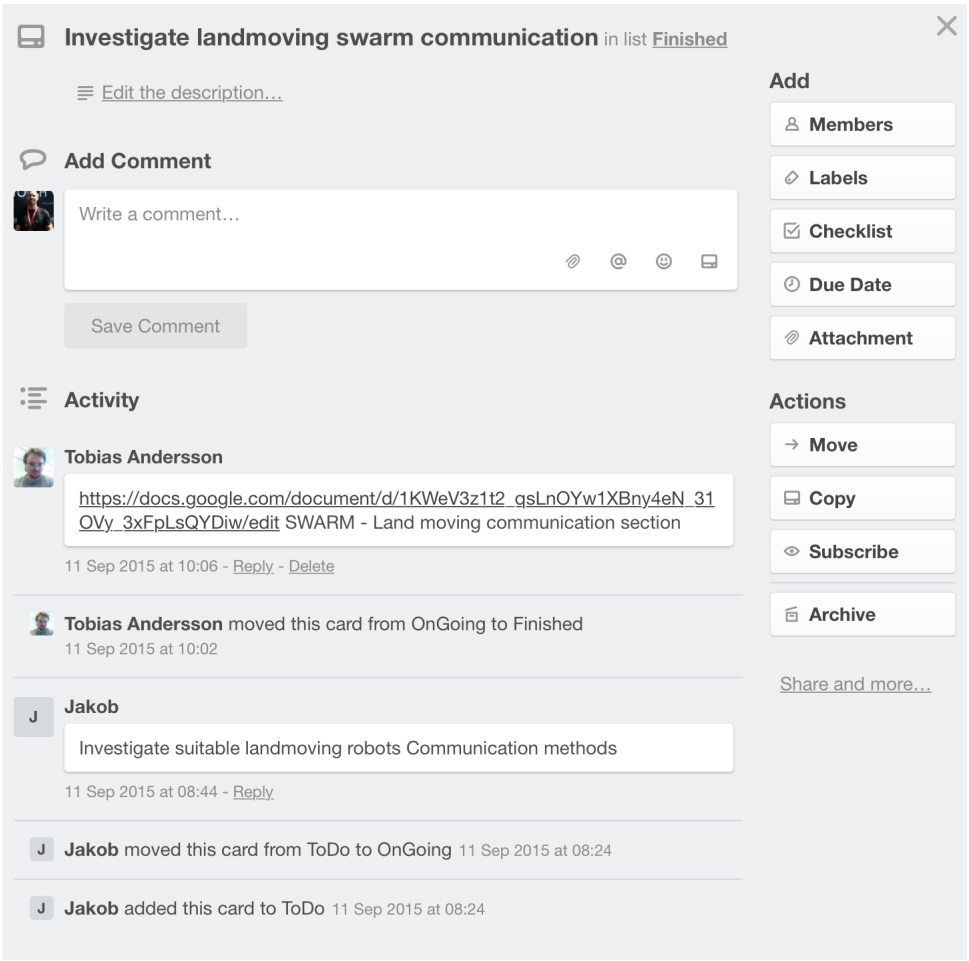
D. Light communication - BOM(Albin Barklund)

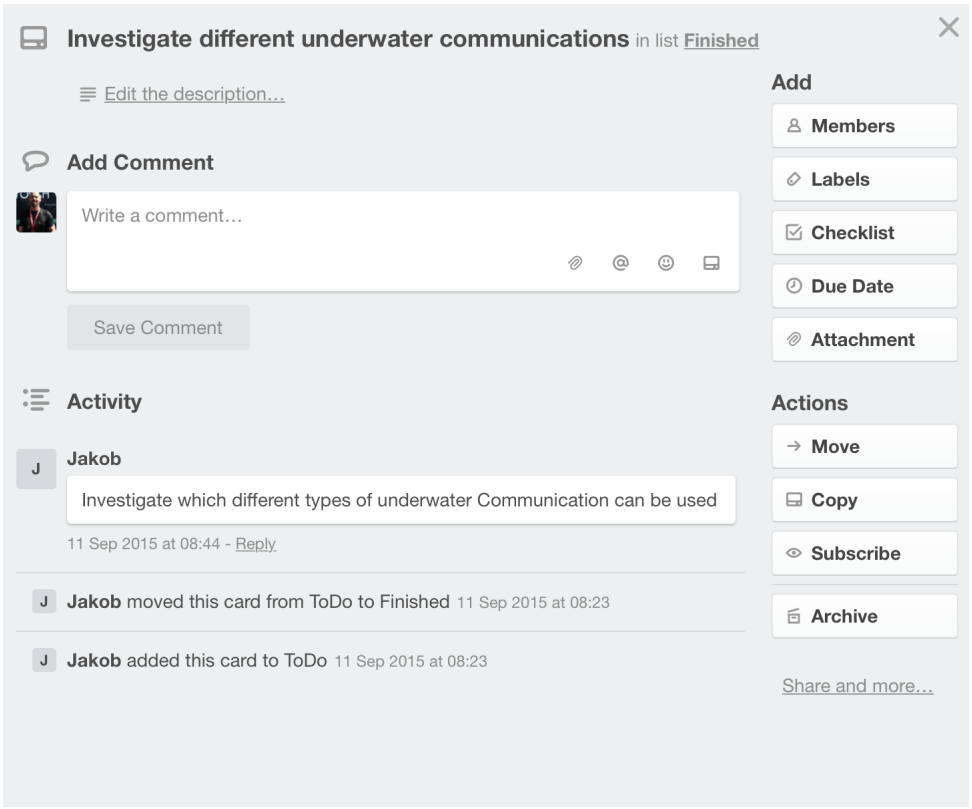
Name	Vendor	Ordernumber
Blue LED	DigiKey	XPCBLU-L1-R250-00V01TR-ND
Photodiode Blue	DigiKey	SD019-141-411-BCT-ND

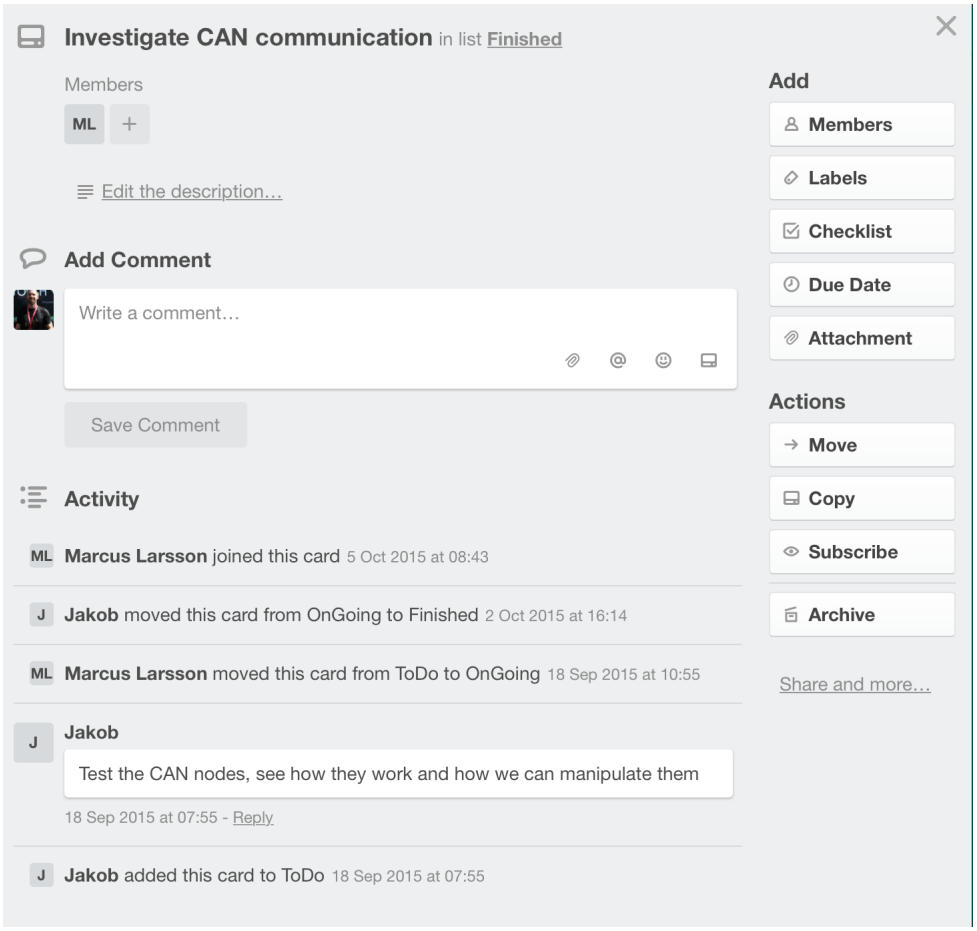
E. Trello Cards(Peter Cederblad)

E.1. Comunication trello Cards

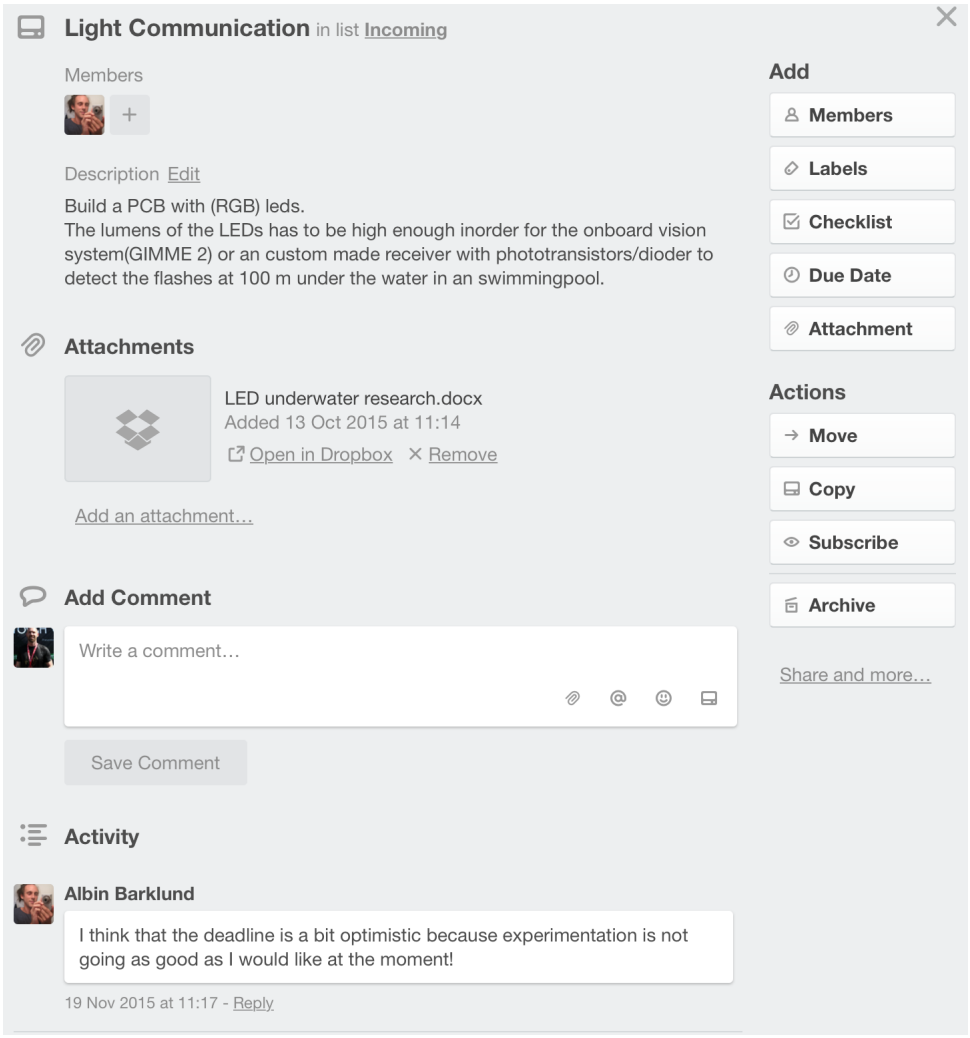


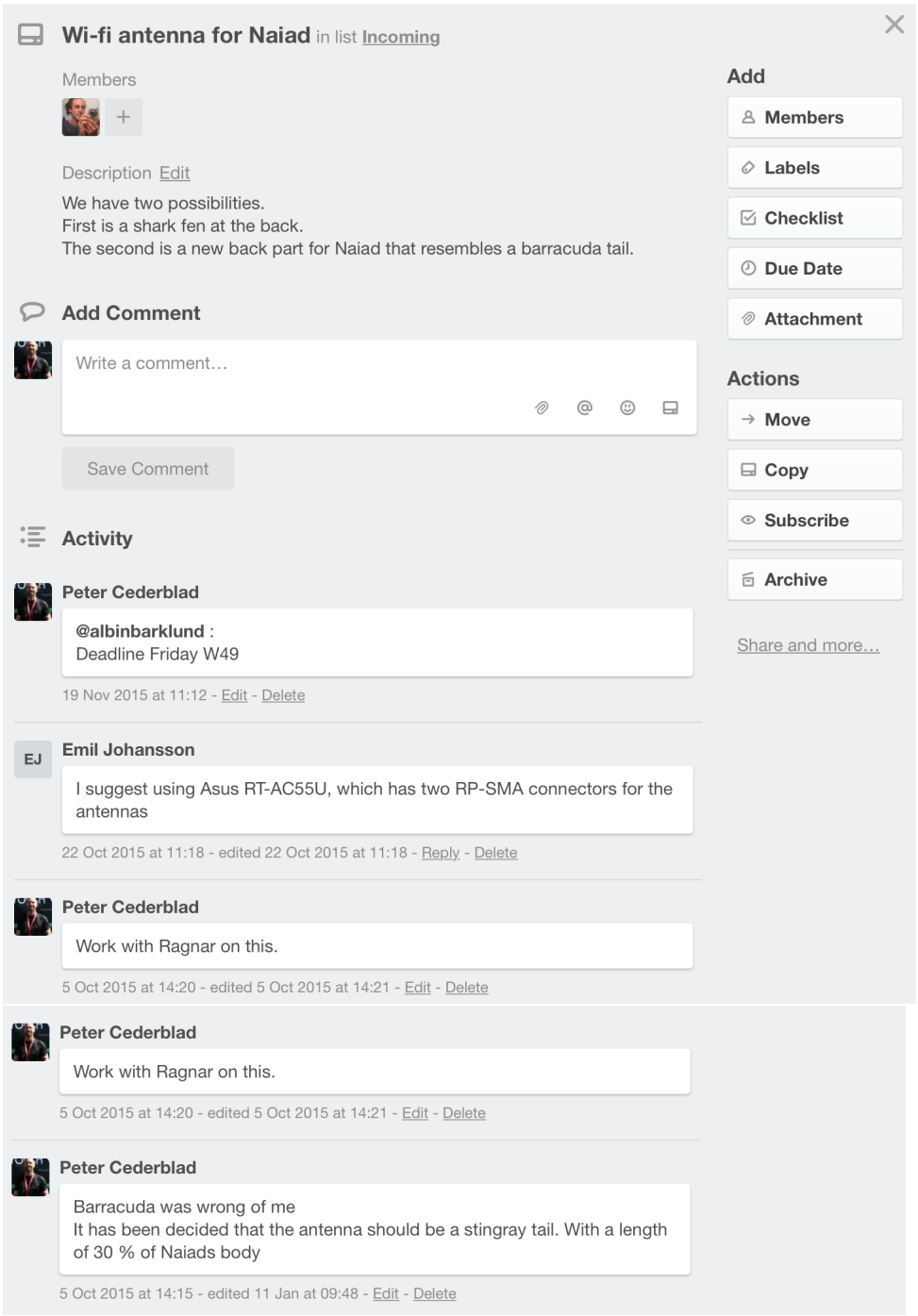

Investigate landmoving swarm communication in list Finished
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Tobias Andersson
https://docs.google.com/document/d/1KWeV3z1t2_qsLnOYw1XBny4eN_31OVy_3xFpLsQYDiw/edit SWARM - Land moving communication section
11 Sep 2015 at 10:06 - Reply - Delete
Tobias Andersson moved this card from OnGoing to Finished
11 Sep 2015 at 10:02
J
Jakob
Investigate suitable landmoving robots Communication methods
11 Sep 2015 at 08:44 - Reply
J
Jakob moved this card from ToDo to OnGoing 11 Sep 2015 at 08:24
J
Jakob added this card to ToDo 11 Sep 2015 at 08:24
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…


Investigate different underwater communications in list Finished
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Jakob
Investigate which different types of underwater Communication can be used
11 Sep 2015 at 08:44 - Reply
Jakob moved this card from ToDo to Finished 11 Sep 2015 at 08:23
Jakob added this card to ToDo 11 Sep 2015 at 08:23
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

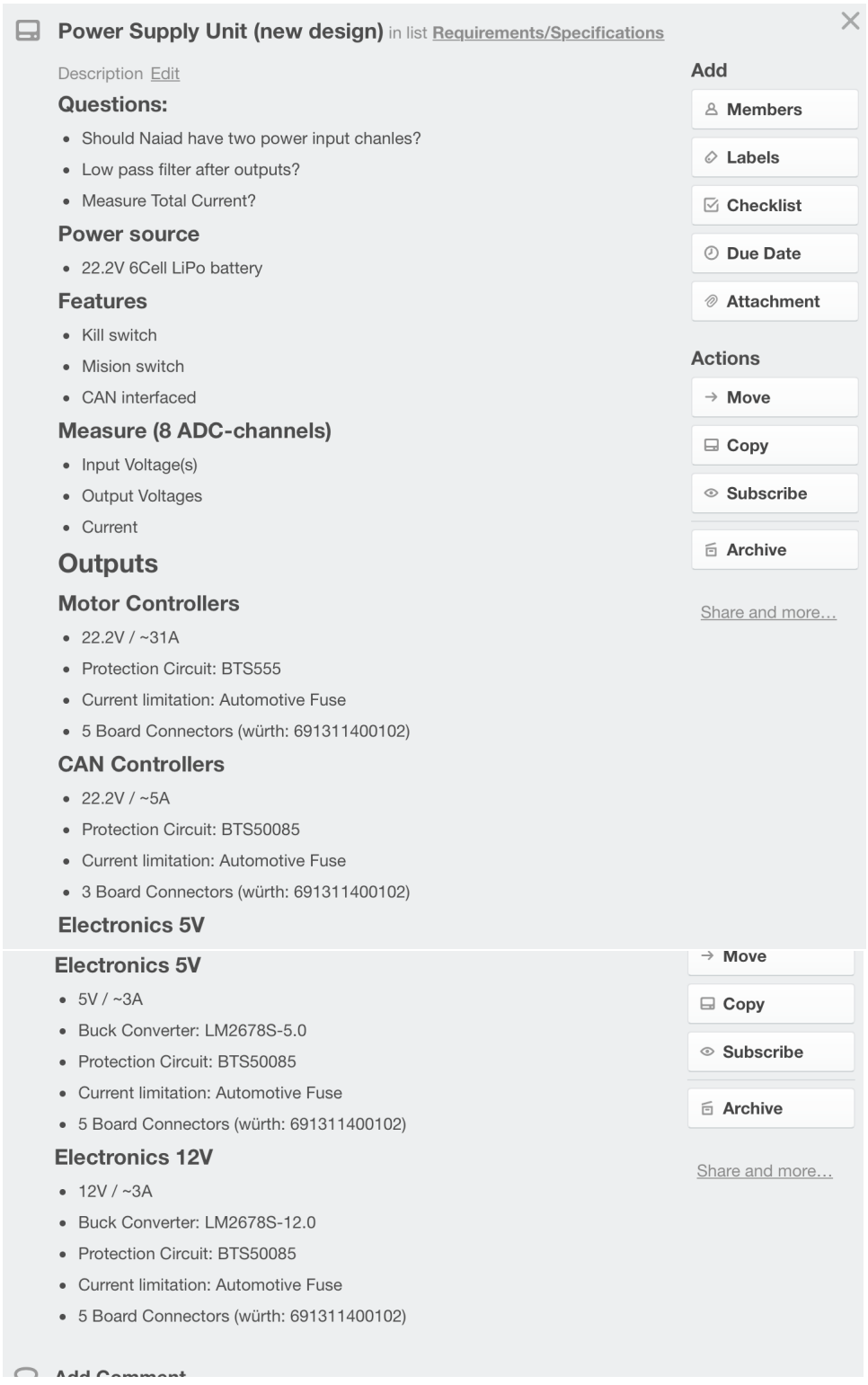

Investigate CAN communication in list Finished
Members
ML
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
ML Marcus Larsson joined this card 5 Oct 2015 at 08:43
J Jakob moved this card from OnGoing to Finished 2 Oct 2015 at 16:14
ML Marcus Larsson moved this card from ToDo to OnGoing 18 Sep 2015 at 10:55
J Jakob
Test the CAN nodes, see how they work and how we can manipulate them
18 Sep 2015 at 07:55 - Reply
J Jakob added this card to ToDo 18 Sep 2015 at 07:55
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

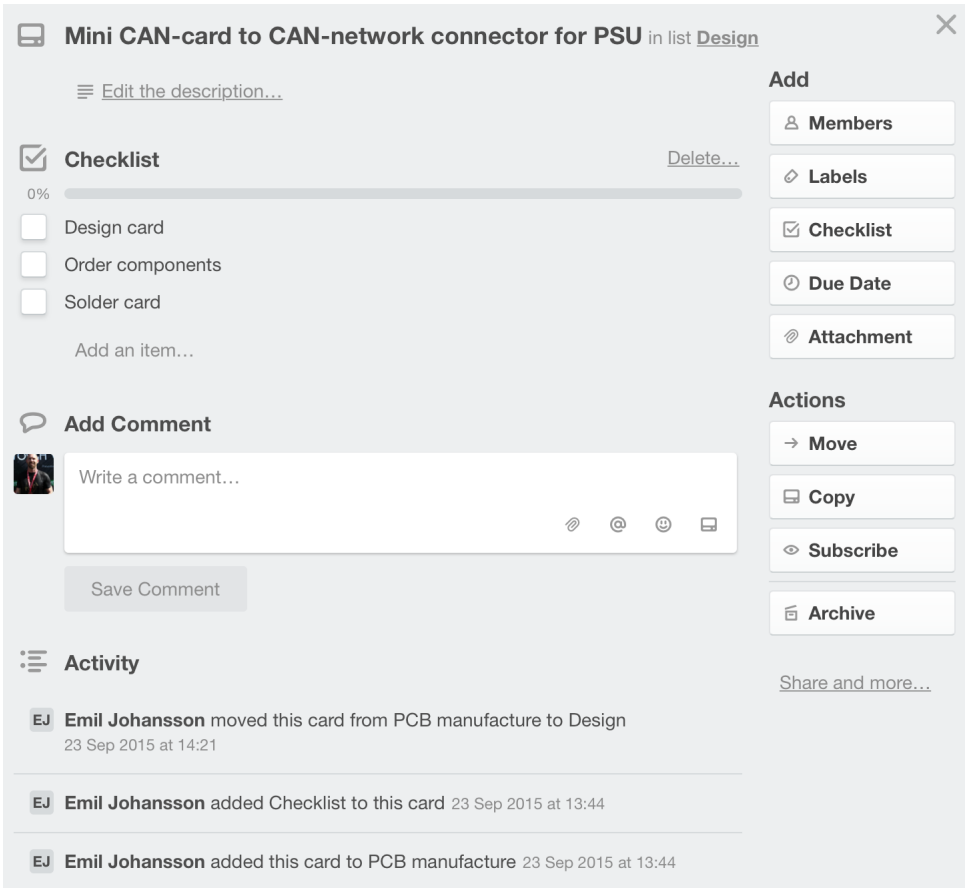
E.2. Hardware trello Cards

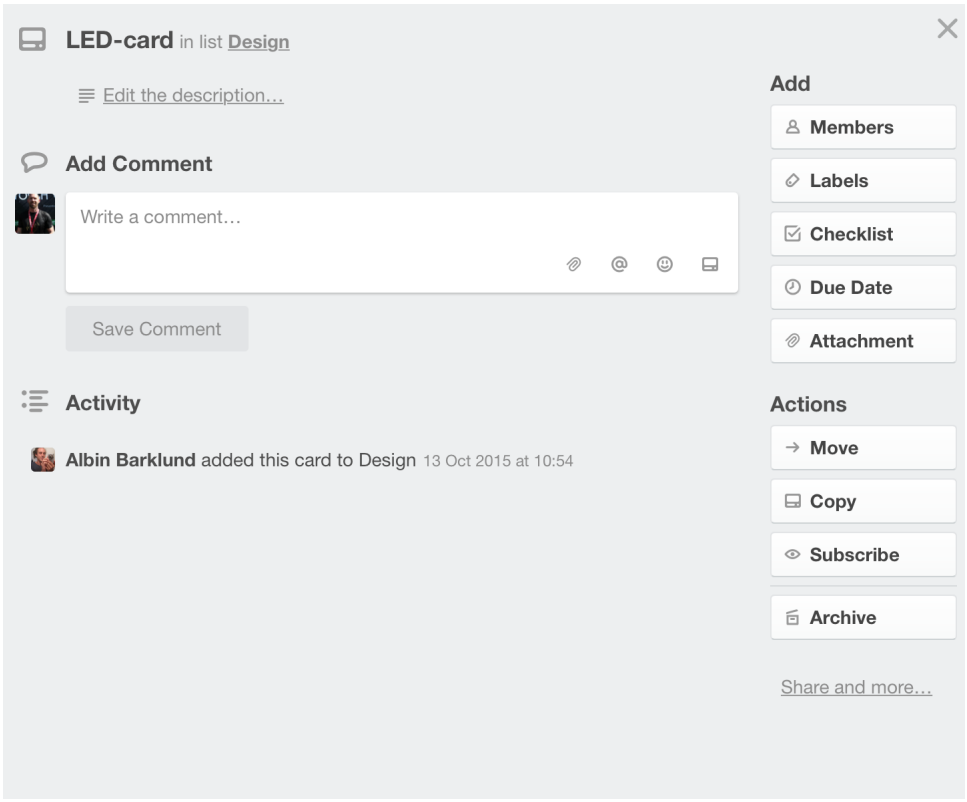


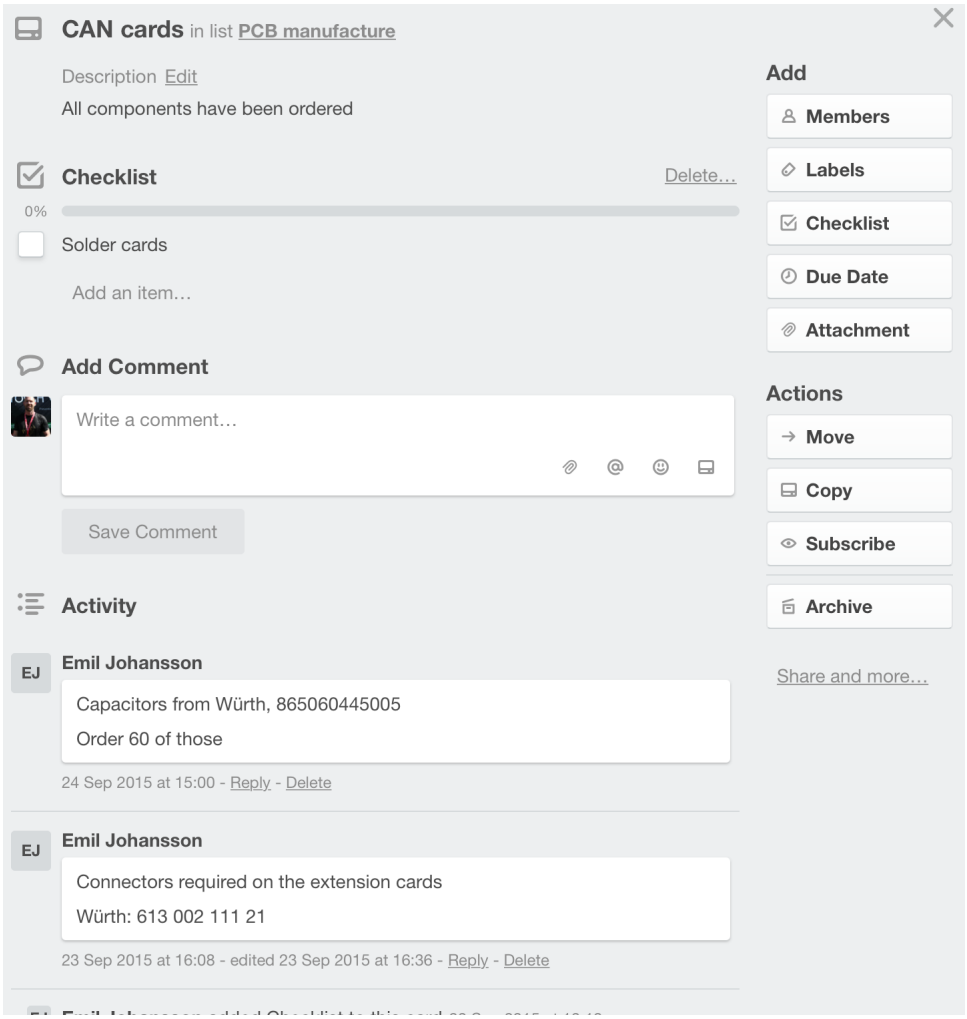


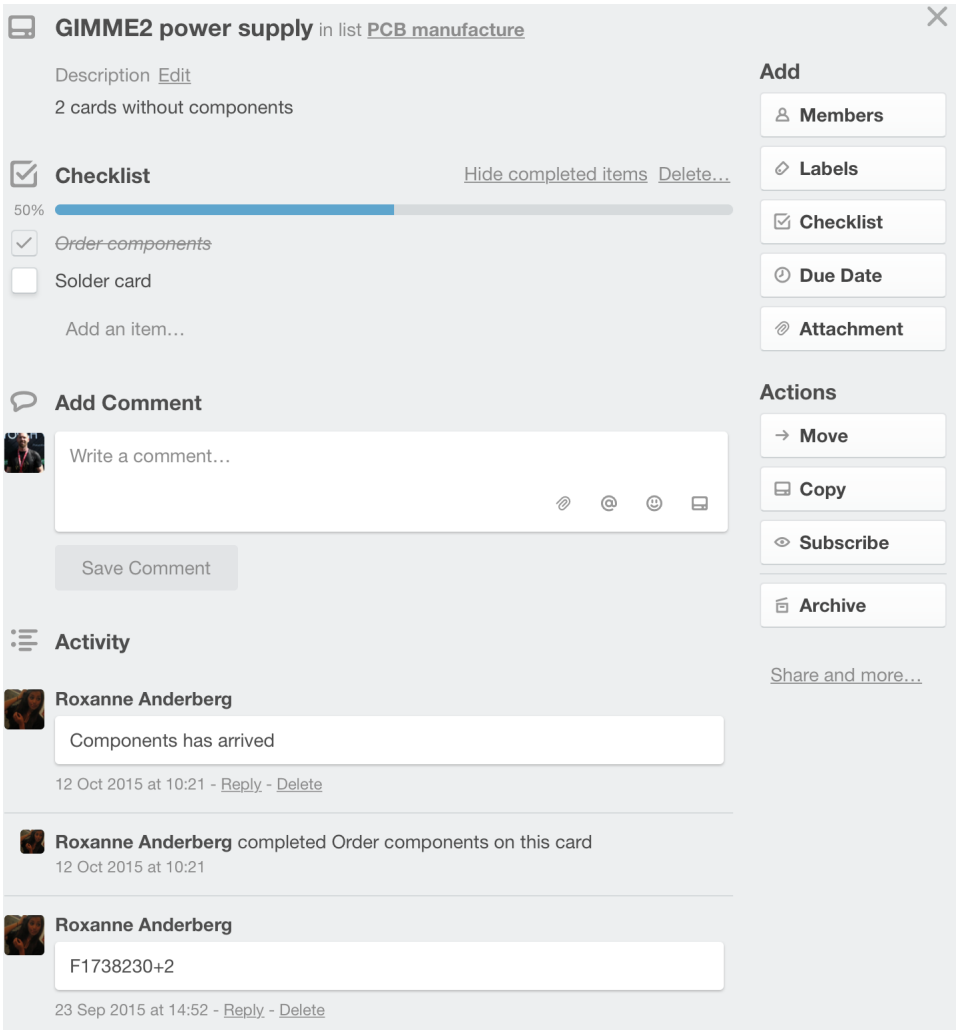
Peter Cederblad
Work with Ragnar on this.
5 Oct 2015 at 14:20 - edited 5 Oct 2015 at 14:21 - Edit - Delete
Peter Cederblad
Barracuda was wrong of me
It has been decided that the antenna should be a stingray tail. With a length of 30 % of Naiads body
5 Oct 2015 at 14:15 - edited 11 Jan at 09:48 - Edit - Delete

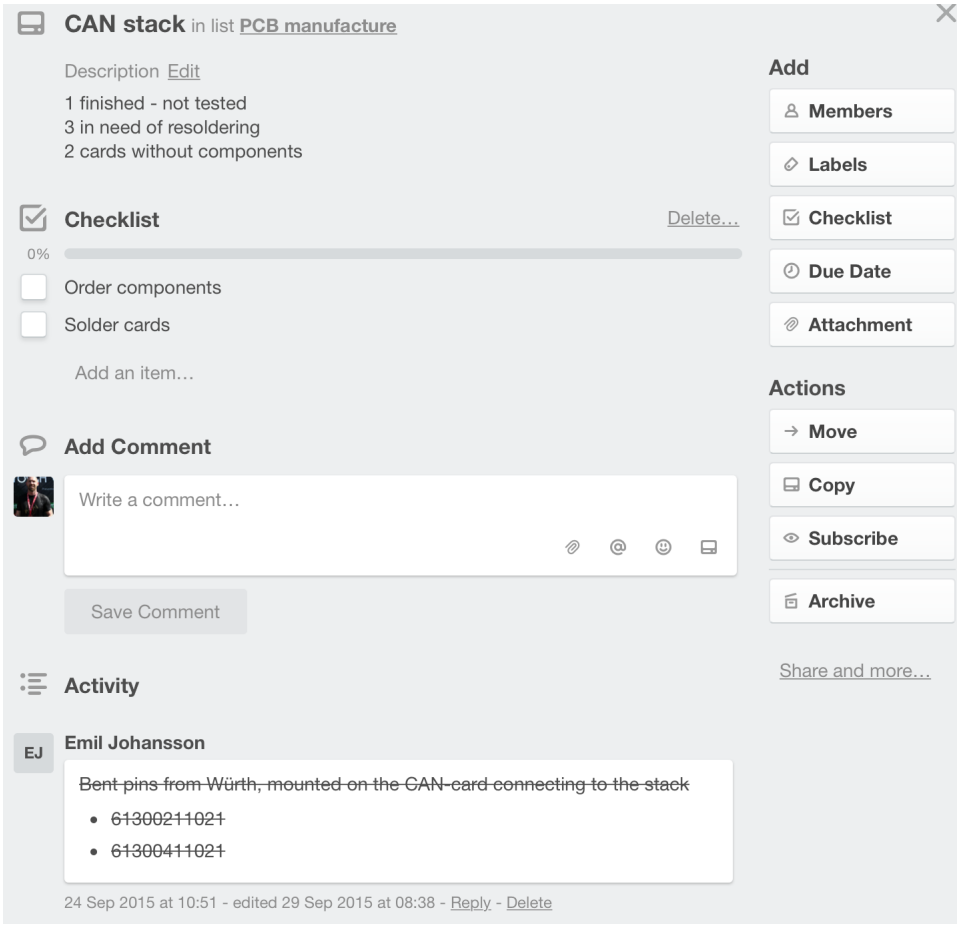

Power Supply Unit (new design) in list Requirements/Specifications
Description Edit
Questions:
Should Naiad have two power input chanles?
Low pass filter after outputs?
Measure Total Current?
Power source
22.2V 6Cell LiPo battery
Features
Kill switch
Mision switch
CAN interfaced
Measure (8 ADC-channels)
Input Voltage(s)
Output Voltages
Current
Outputs
Motor Controllers
22.2V / ~31A
Protection Circuit: BTS555
Current limitation: Automotive Fuse
5 Board Connectors (würth: 691311400102)
CAN Controllers
22.2V / ~5A
Protection Circuit: BTS50085
Current limitation: Automotive Fuse
3 Board Connectors (würth: 691311400102)
Electronics 5V
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…
Electronics 5V
5V / ~3A
Buck Converter: LM2678S-5.0
Protection Circuit: BTS50085
Current limitation: Automotive Fuse
5 Board Connectors (würth: 691311400102)
Electronics 12V
12V / ~3A
Buck Converter: LM2678S-12.0
Protection Circuit: BTS50085
Current limitation: Automotive Fuse
5 Board Connectors (würth: 691311400102)
Move
Copy
Subscribe
Archive
Share and more…

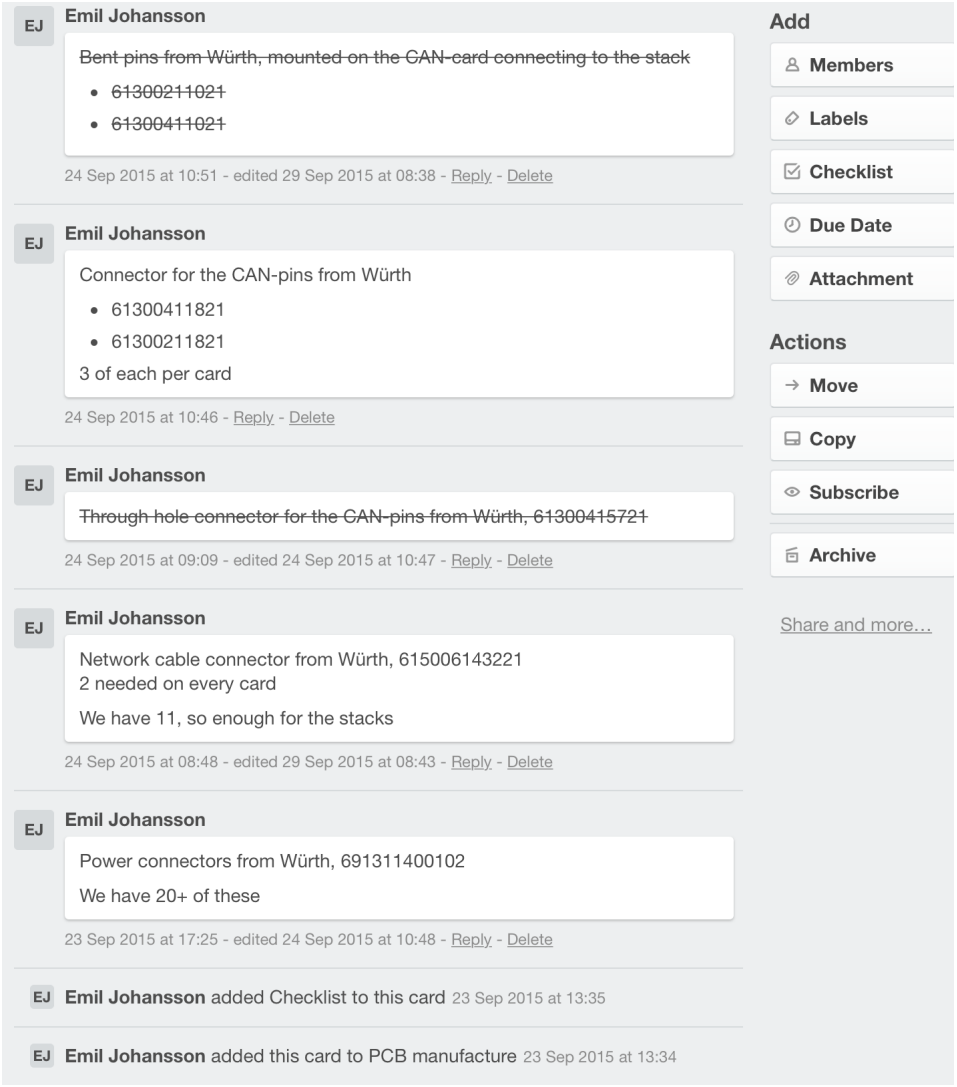

Mini CAN-card to CAN-network connector for PSU in list Design
Edit the description…
Checklist
Delete…
0%
Design card
Order components
Solder card
Add an item…
Add Comment
Write a comment…
Save Comment
Activity
EJ Emil Johansson moved this card from PCB manufacture to Design
23 Sep 2015 at 14:21
EJ Emil Johansson added Checklist to this card 23 Sep 2015 at 13:44
EJ Emil Johansson added this card to PCB manufacture 23 Sep 2015 at 13:44
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

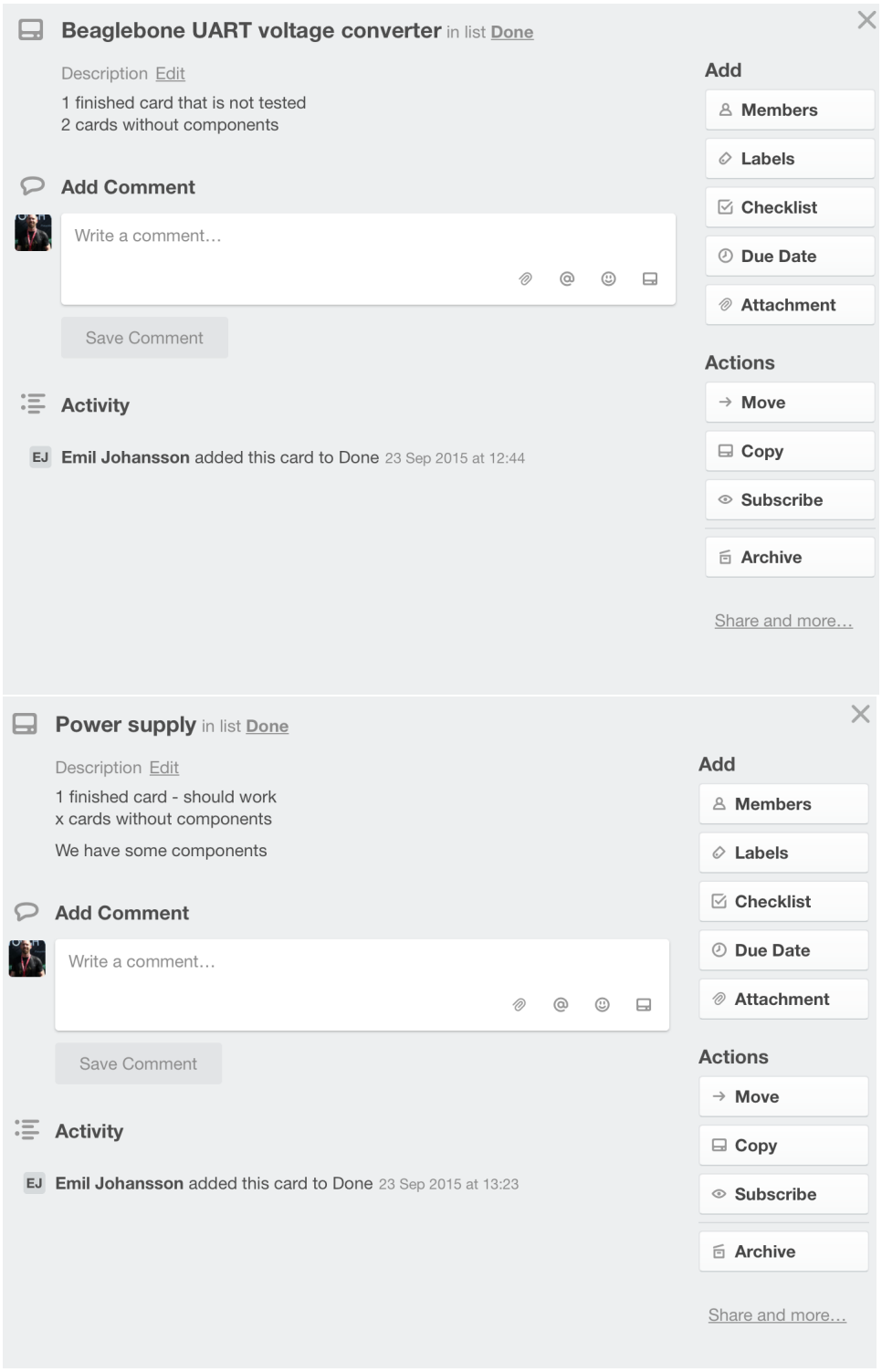

LED-card in list Design
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Albin Barklund added this card to Design 13 Oct 2015 at 10:54
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

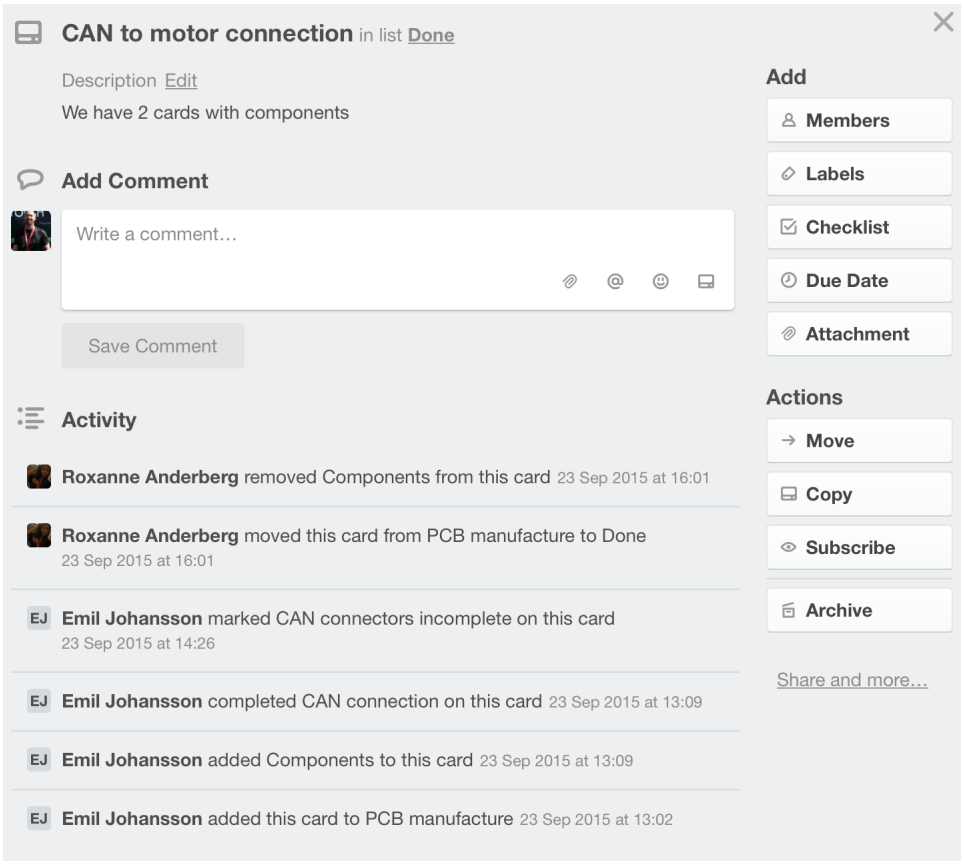

CAN cards in list PCB manufacture
Description Edit
All components have been ordered
Checklist
Delete…
0%
Solder cards
Add an item…
Add Comment
Write a comment…
Save Comment
Activity
EJ
Emil Johansson
Capacitors from Würth, 865060445005
Order 60 of those
24 Sep 2015 at 15:00 - Reply - Delete
EJ
Emil Johansson
Connectors required on the extension cards
Würth: 613 002 111 21
23 Sep 2015 at 16:08 - edited 23 Sep 2015 at 16:36 - Reply - Delete
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

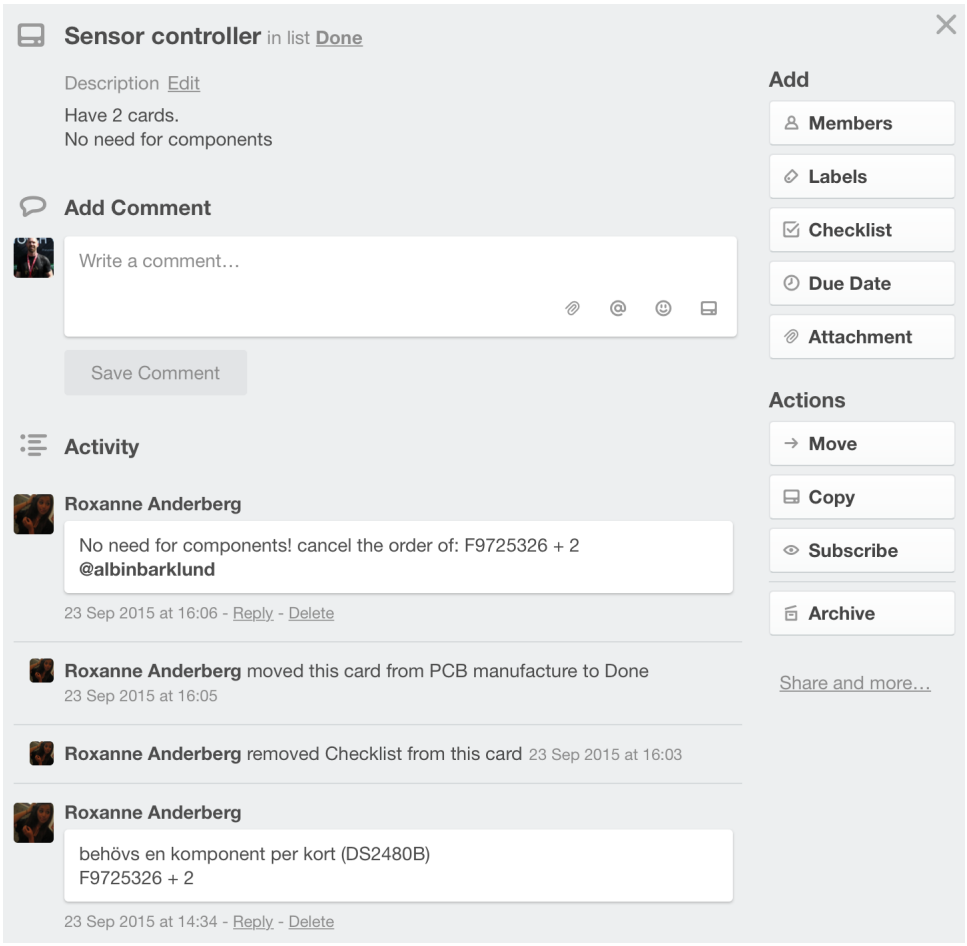

GIMME2 power supply in list PCB manufacture
Description Edit
2 cards without components
Checklist
Hide completed items Delete...
50%
Order components
Solder card
Add an item...
Add Comment
Write a comment...
Save Comment
Activity
Roxanne Anderberg
Components has arrived
12 Oct 2015 at 10:21 - Reply - Delete
Roxanne Anderberg completed Order components on this card
12 Oct 2015 at 10:21
Roxanne Anderberg
F1738230+2
23 Sep 2015 at 14:52 - Reply - Delete
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more...

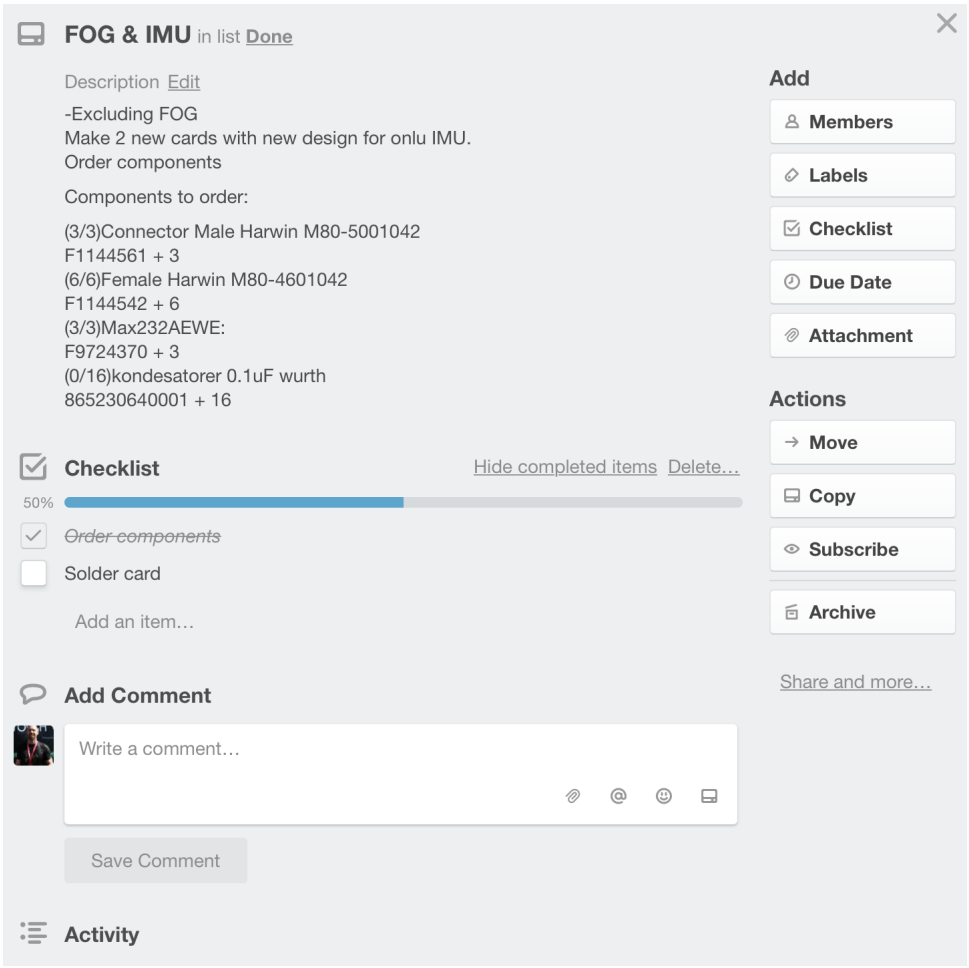

CAN stack in list PCB manufacture
Description Edit
1 finished - not tested
3 in need of resoldering
2 cards without components
Checklist
Delete…
0%
Order components
Solder cards
Add an item…
Add Comment
Write a comment…
Save Comment
Activity
EJ
Emil Johansson
Bent pins from Würth, mounted on the CAN-card connecting to the stack
61300211021
61300411021
24 Sep 2015 at 10:51 - edited 29 Sep 2015 at 08:38 - Reply - Delete
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

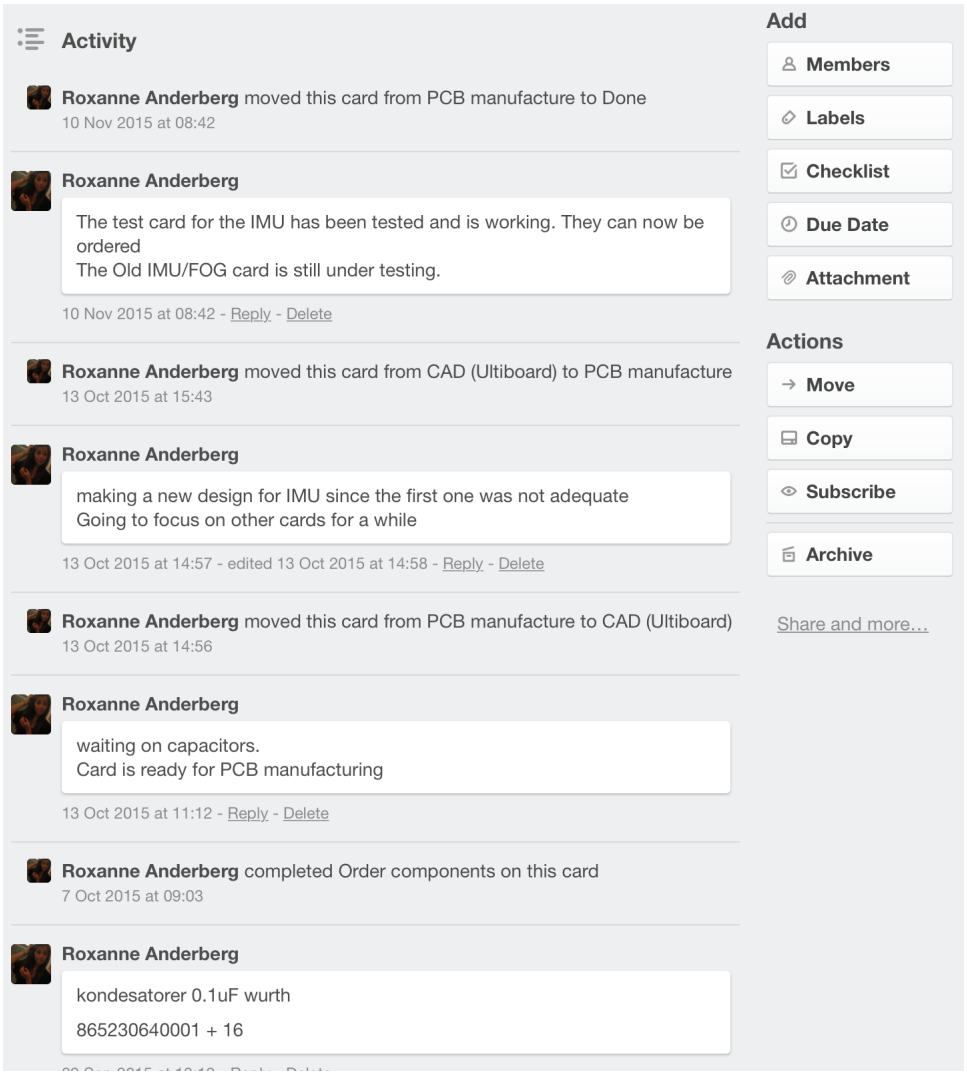

EJ
Emil Johansson
~~Bent pins from Würth, mounted on the CAN-card connecting to the stack~~
- ~~61300211021~~
- ~~61300411021~~
24 Sep 2015 at 10:51 - edited 29 Sep 2015 at 08:38 - Reply - Delete
EJ
Emil Johansson
Connector for the CAN-pins from Würth
- 61300411821
- 61300211821
3 of each per card
24 Sep 2015 at 10:46 - Reply - Delete
EJ
Emil Johansson
~~Through hole connector for the CAN-pins from Würth, 61300415721~~
24 Sep 2015 at 09:09 - edited 24 Sep 2015 at 10:47 - Reply - Delete
EJ
Emil Johansson
Network cable connector from Würth, 615006143221
2 needed on every card
We have 11, so enough for the stacks
24 Sep 2015 at 08:48 - edited 29 Sep 2015 at 08:43 - Reply - Delete
EJ
Emil Johansson
Power connectors from Würth, 691311400102
We have 20+ of these
23 Sep 2015 at 17:25 - edited 24 Sep 2015 at 10:48 - Reply - Delete
EJ Emil Johansson added Checklist to this card 23 Sep 2015 at 13:35
EJ Emil Johansson added this card to PCB manufacture 23 Sep 2015 at 13:34
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

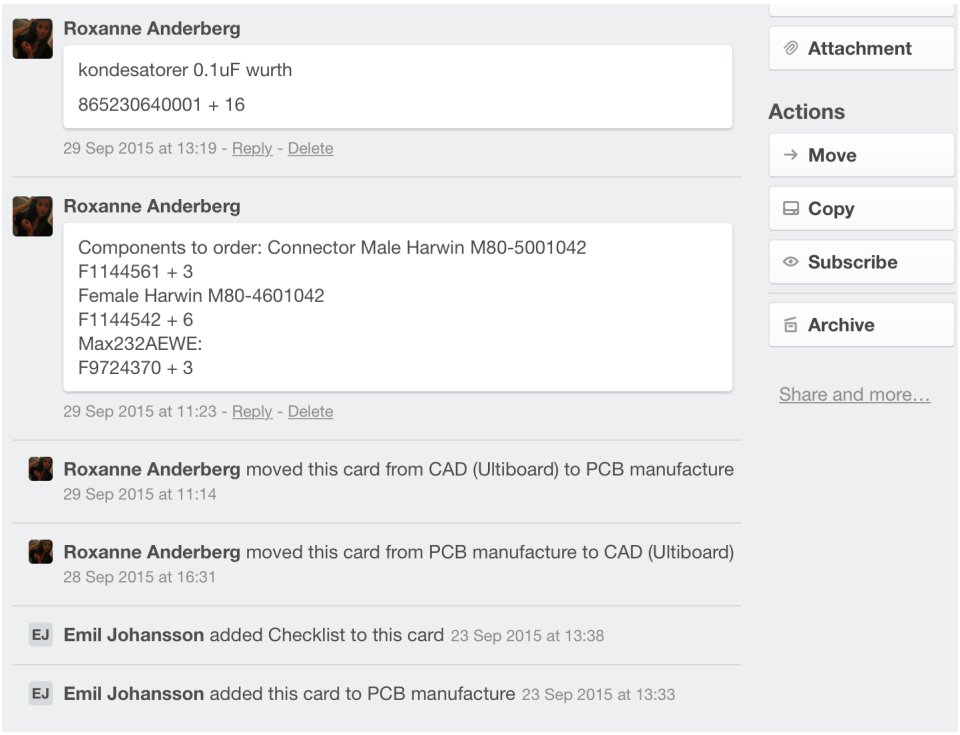

Beaglebone UART voltage converter in list Done
Description Edit
1 finished card that is not tested
2 cards without components
Add Comment
Write a comment…
Save Comment
Activity
EJ Emil Johansson added this card to Done 23 Sep 2015 at 12:44
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…
Power supply in list Done
Description Edit
1 finished card - should work
x cards without components
We have some components
Add Comment
Write a comment…
Save Comment
Activity
EJ Emil Johansson added this card to Done 23 Sep 2015 at 13:23
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

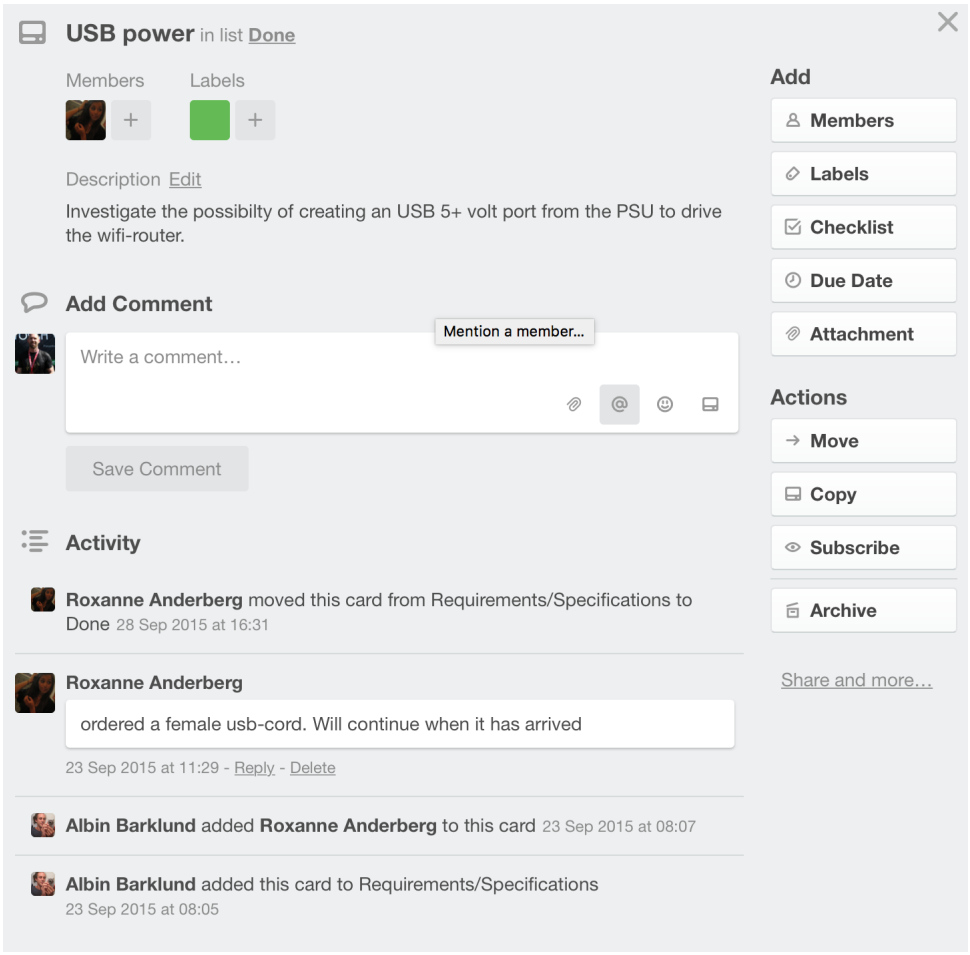

CAN to motor connection in list Done
Description Edit
We have 2 cards with components
Add Comment
Write a comment…
Save Comment
Activity
Roxanne Anderberg removed Components from this card 23 Sep 2015 at 16:01
Roxanne Anderberg moved this card from PCB manufacture to Done
23 Sep 2015 at 16:01
EJ Emil Johansson marked CAN connectors incomplete on this card
23 Sep 2015 at 14:26
EJ Emil Johansson completed CAN connection on this card 23 Sep 2015 at 13:09
EJ Emil Johansson added Components to this card 23 Sep 2015 at 13:09
EJ Emil Johansson added this card to PCB manufacture 23 Sep 2015 at 13:02
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…


Sensor controller in list Done
Description Edit
Have 2 cards.
No need for components
Add Comment
Write a comment…
Save Comment
Activity
Roxanne Anderberg
No need for components! cancel the order of: F9725326 + 2
@albinbarklund
23 Sep 2015 at 16:06 - Reply - Delete
Roxanne Anderberg moved this card from PCB manufacture to Done
23 Sep 2015 at 16:05
Roxanne Anderberg removed Checklist from this card 23 Sep 2015 at 16:03
Roxanne Anderberg
behövs en komponent per kort (DS2480B)
F9725326 + 2
23 Sep 2015 at 14:34 - Reply - Delete
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

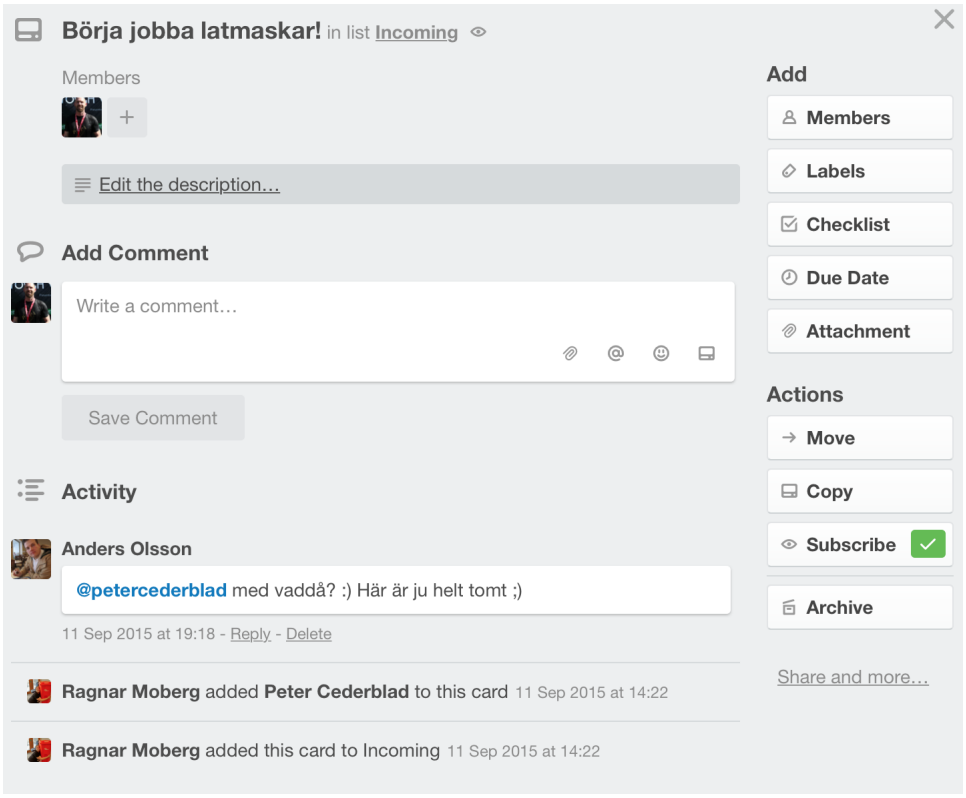

FOG & IMU in list Done
Description Edit
-Excluding FOG
Make 2 new cards with new design for onlu IMU.
Order components
Components to order:
(3/3)Connector Male Harwin M80-5001042
F1144561 + 3
(6/6)Female Harwin M80-4601042
F1144542 + 6
(3/3)Max232AEWE:
F9724370 + 3
(0/16)kondesatorer 0.1uF wurth
865230640001 + 16
Checklist
Hide completed items
Delete…
50%
Order components
Solder card
Add an item…
Add Comment
Write a comment…
Save Comment
Activity
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

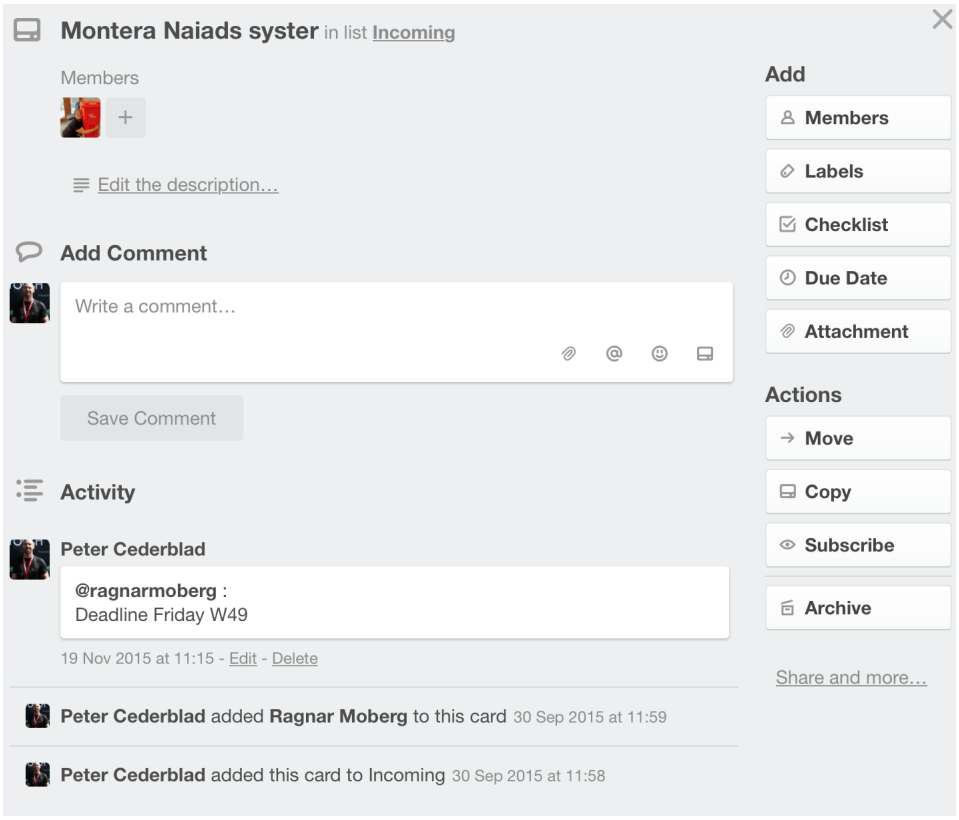

Activity
Roxanne Anderberg moved this card from PCB manufacture to Done
10 Nov 2015 at 08:42
Roxanne Anderberg
The test card for the IMU has been tested and is working. They can now be ordered
The Old IMU/FOG card is still under testing.
10 Nov 2015 at 08:42 - Reply - Delete
Roxanne Anderberg moved this card from CAD (Ultiboard) to PCB manufacture
13 Oct 2015 at 15:43
Roxanne Anderberg
making a new design for IMU since the first one was not adequate
Going to focus on other cards for a while
13 Oct 2015 at 14:57 - edited 13 Oct 2015 at 14:58 - Reply - Delete
Roxanne Anderberg moved this card from PCB manufacture to CAD (Ultiboard)
13 Oct 2015 at 14:56
Roxanne Anderberg
waiting on capacitors.
Card is ready for PCB manufacturing
13 Oct 2015 at 11:12 - Reply - Delete
Roxanne Anderberg completed Order components on this card
7 Oct 2015 at 09:03
Roxanne Anderberg
kondesatorer 0.1uF wurth
865230640001 + 16
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more...

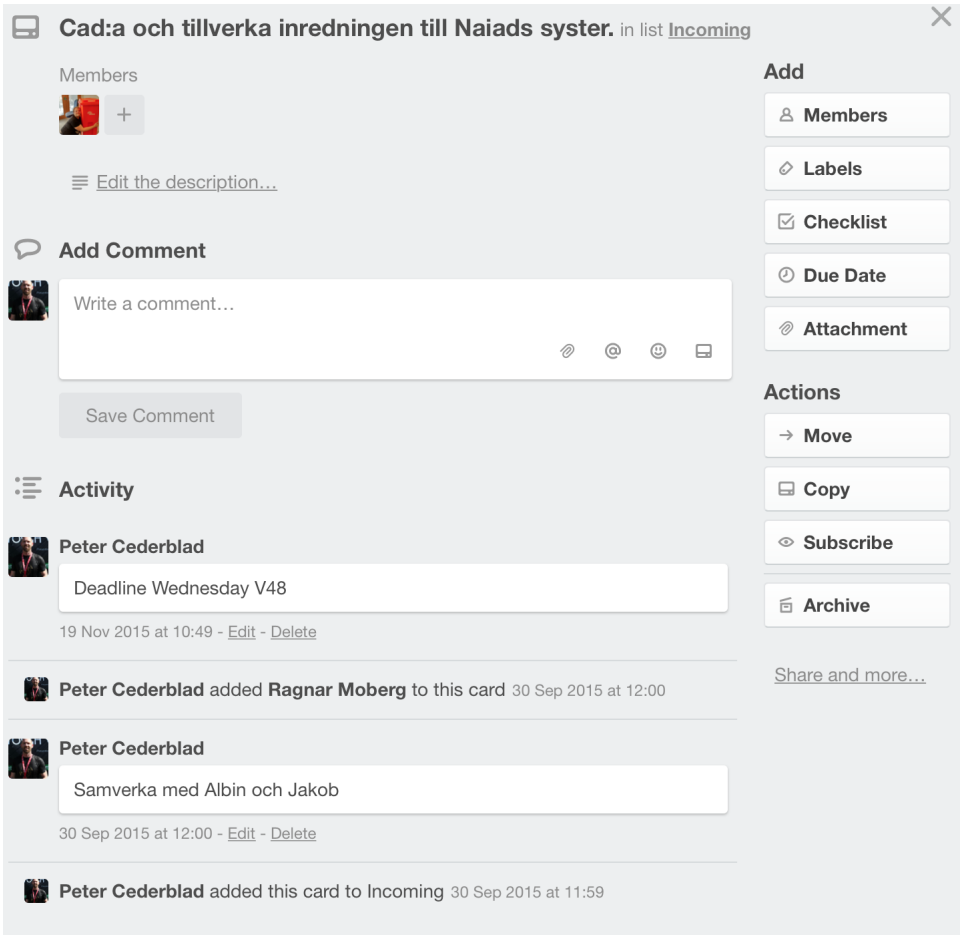

Roxanne Anderberg
kondesatorer 0.1uF wurth
865230640001 + 16
29 Sep 2015 at 13:19 - Reply - Delete
Roxanne Anderberg
Components to order: Connector Male Harwin M80-5001042
F1144561 + 3
Female Harwin M80-4601042
F1144542 + 6
Max232AEWE:
F9724370 + 3
29 Sep 2015 at 11:23 - Reply - Delete
Roxanne Anderberg moved this card from CAD (Ultiboard) to PCB manufacture
29 Sep 2015 at 11:14
Roxanne Anderberg moved this card from PCB manufacture to CAD (Ultiboard)
28 Sep 2015 at 16:31
EJ Emil Johansson added Checklist to this card 23 Sep 2015 at 13:38
EJ Emil Johansson added this card to PCB manufacture 23 Sep 2015 at 13:33
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

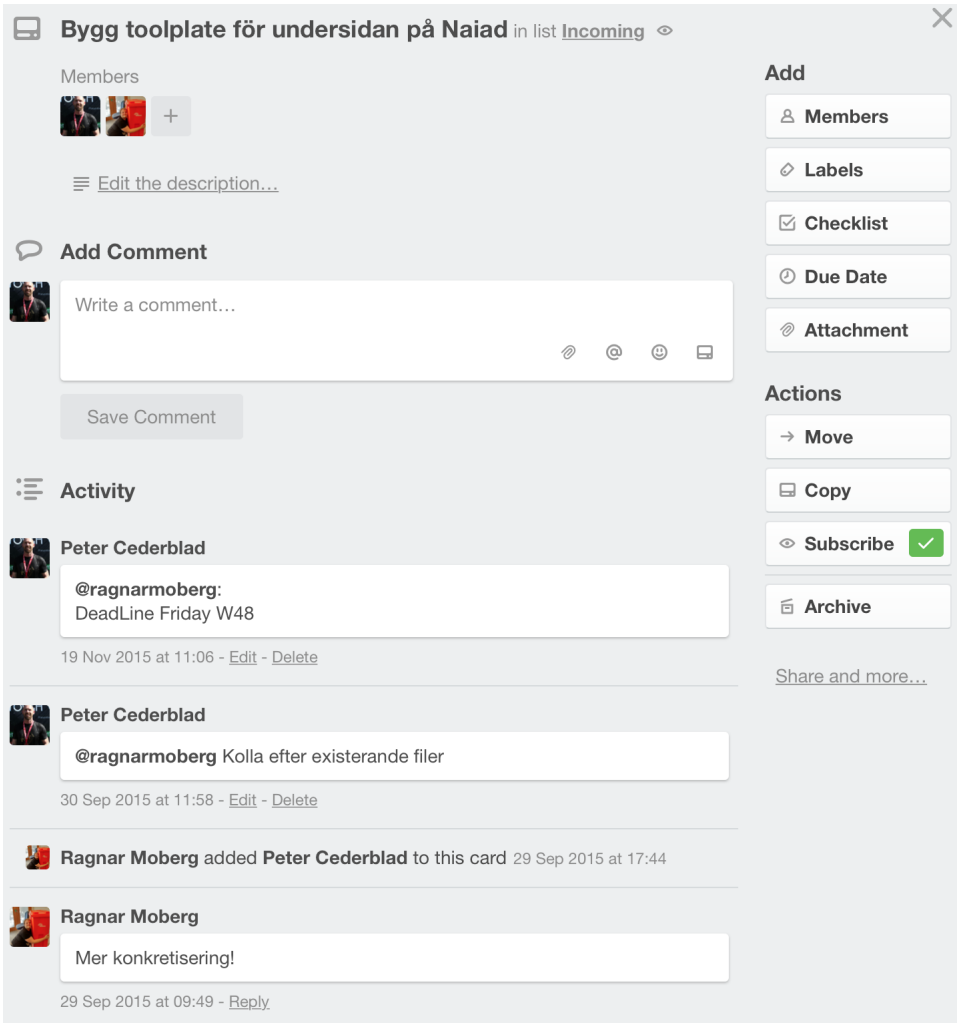

USB power in list Done
Members
Labels
Description Edit
Investigate the possibilty of creating an USB 5+ volt port from the PSU to drive the wifi-router.
Add Comment
Mention a member...
Write a comment…
Save Comment
Activity
Roxanne Anderberg moved this card from Requirements/Specifications to Done 28 Sep 2015 at 16:31
Roxanne Anderberg
ordered a female usb-cord. Will continue when it has arrived
23 Sep 2015 at 11:29 - Reply - Delete
Albin Barklund added Roxanne Anderberg to this card 23 Sep 2015 at 08:07
Albin Barklund added this card to Requirements/Specifications
23 Sep 2015 at 08:05
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

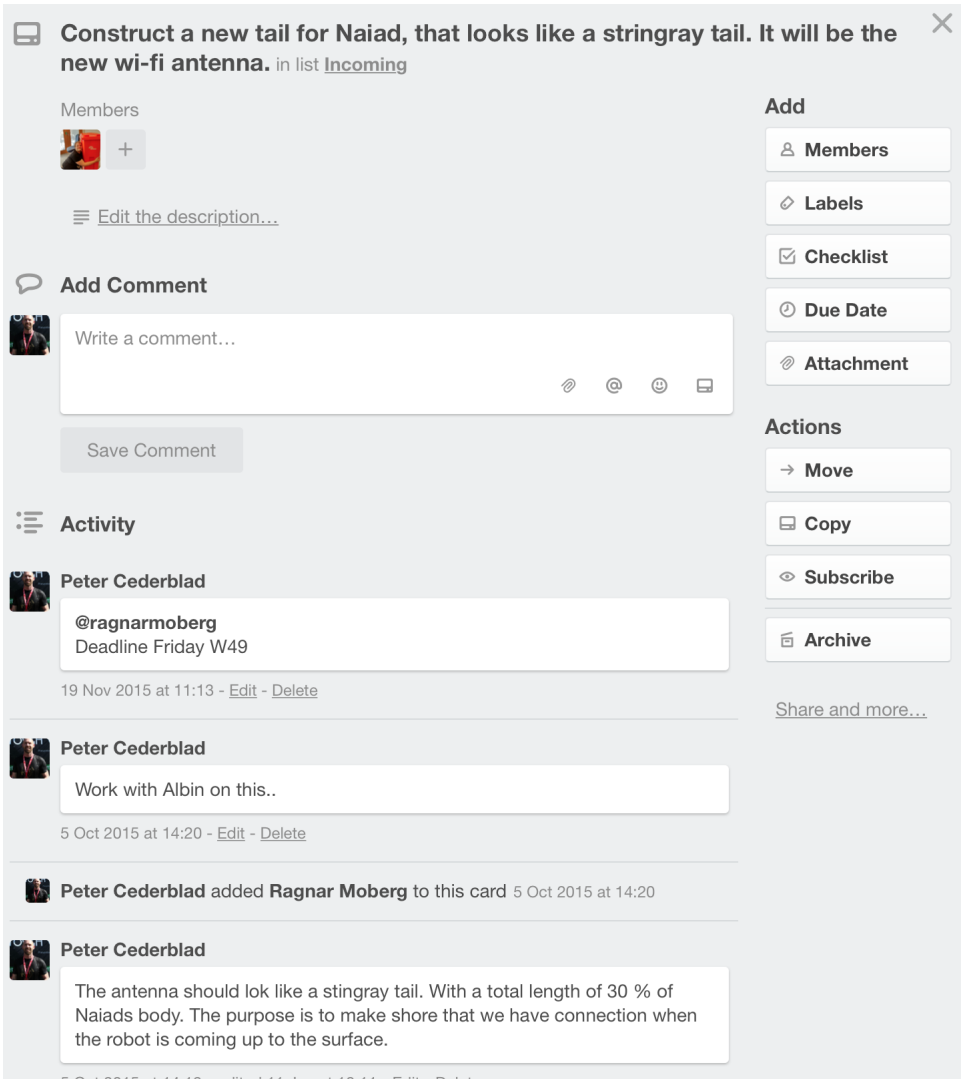
E.3. Mechanics trello Cards

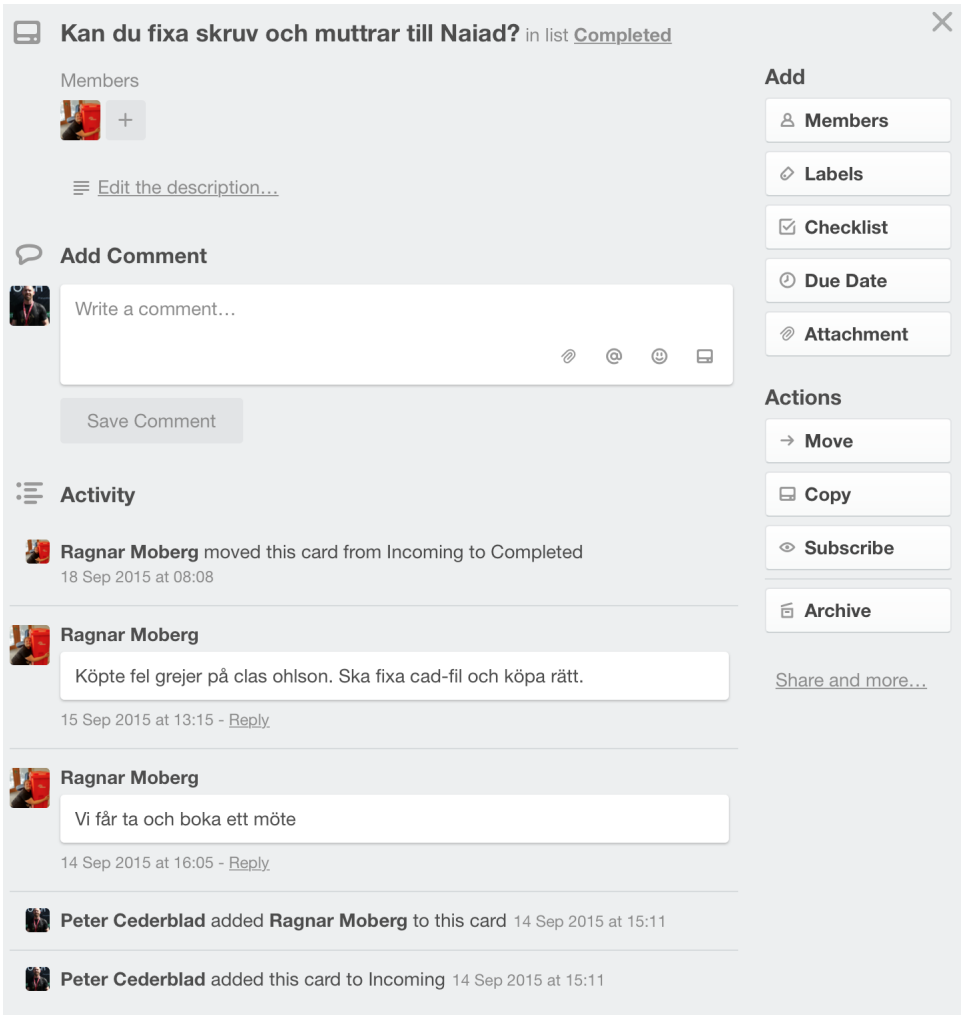



Montera Naiads syster in list Incoming
Members
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Peter Cederblad
@ragnarmoberg :
Deadline Friday W49
19 Nov 2015 at 11:15 - Edit - Delete
Peter Cederblad added Ragnar Moberg to this card 30 Sep 2015 at 11:59
Peter Cederblad added this card to Incoming 30 Sep 2015 at 11:58
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

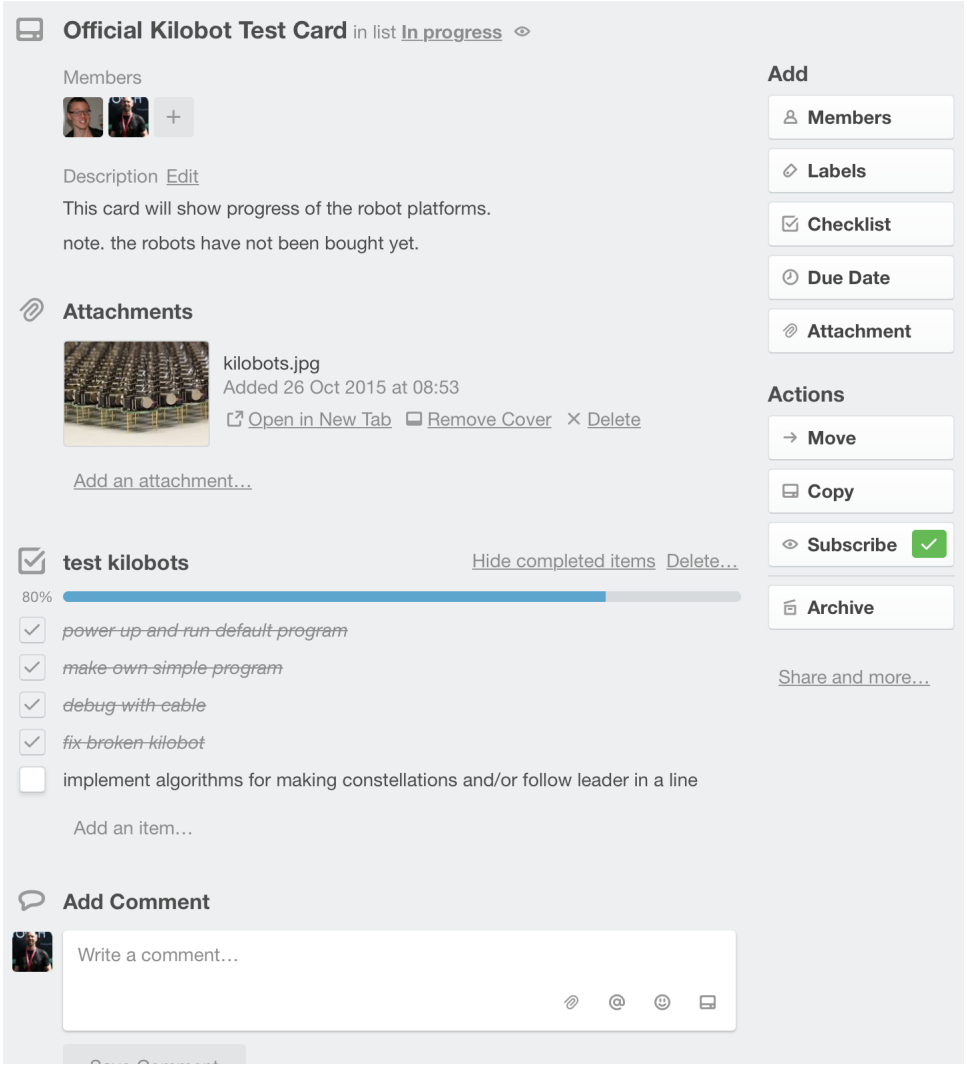

Cad:a och tillverka inredningen till Naiads syster. in list Incoming
Members
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Peter Cederblad
Deadline Wednesday V48
19 Nov 2015 at 10:49 - Edit - Delete
Peter Cederblad added Ragnar Moberg to this card 30 Sep 2015 at 12:00
Peter Cederblad
Samverka med Albin och Jakob
30 Sep 2015 at 12:00 - Edit - Delete
Peter Cederblad added this card to Incoming 30 Sep 2015 at 11:59
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…


Bygg toolplate för undersidan på Naiad in list Incoming
Members
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Peter Cederblad
@ragnarmoberg:
DeadLine Friday W48
19 Nov 2015 at 11:06 - Edit - Delete
Peter Cederblad
@ragnarmoberg Kolla efter existerande filer
30 Sep 2015 at 11:58 - Edit - Delete
Ragnar Moberg added Peter Cederblad to this card 29 Sep 2015 at 17:44
Ragnar Moberg
Mer konkretisering!
29 Sep 2015 at 09:49 - Reply
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…


Construct a new tail for Naiad, that looks like a stringray tail. It will be the new wi-fi antenna. in list Incoming
Members
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Peter Cederblad
@ragnarmoberg
Deadline Friday W49
19 Nov 2015 at 11:13 - Edit - Delete
Peter Cederblad
Work with Albin on this..
5 Oct 2015 at 14:20 - Edit - Delete
Peter Cederblad added Ragnar Moberg to this card 5 Oct 2015 at 14:20
Peter Cederblad
The antenna should lok like a stingray tail. With a total length of 30 % of Naiads body. The purpose is to make shore that we have connection when the robot is coming up to the surface.
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…


Kan du fixa skruv och muttrar till Naiad? in list Completed
Members
Edit the description…
Add Comment
Write a comment…
Save Comment
Activity
Ragnar Moberg moved this card from Incoming to Completed
18 Sep 2015 at 08:08
Ragnar Moberg
Köpte fel grejer på clas ohlson. Ska fixa cad-fil och köpa rätt.
15 Sep 2015 at 13:15 - Reply
Ragnar Moberg
Vi får ta och boka ett möte
14 Sep 2015 at 16:05 - Reply
Peter Cederblad added Ragnar Moberg to this card 14 Sep 2015 at 15:11
Peter Cederblad added this card to Incoming 14 Sep 2015 at 15:11
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

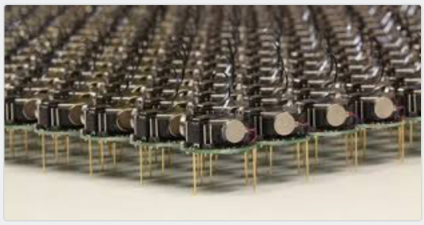
E.4. Software trello Cards



Activity

Peter Cederblad completed fix broken kilobot on this card 24 Nov 2015 at 12:09

Jonatan Tidare attached kilobots.jpg to this card



26 Oct 2015 at 08:53 - Reply

Jonatan Tidare added **Peter Cederblad** to this card 26 Oct 2015 at 08:52

Jonatan Tidare joined this card 26 Oct 2015 at 08:52

Jonatan Tidare completed debug with cable on this card 26 Oct 2015 at 08:52

Jonatan Tidare completed make own simple program on this card
26 Oct 2015 at 08:52

Jonatan Tidare completed power up and run default program on this card
26 Oct 2015 at 08:52

Jonatan Tidare added test kilobots to this card 26 Oct 2015 at 08:50

Jonatan Tidare moved this card from Todo to In progress 26 Oct 2015 at 08:50

Jonatan Tidare

We will continue on the platform testing when we have the robots.

29 Sep 2015 at 08:32 - Reply

Add

Members

Labels

Checklist

Due Date

Attachment

Actions

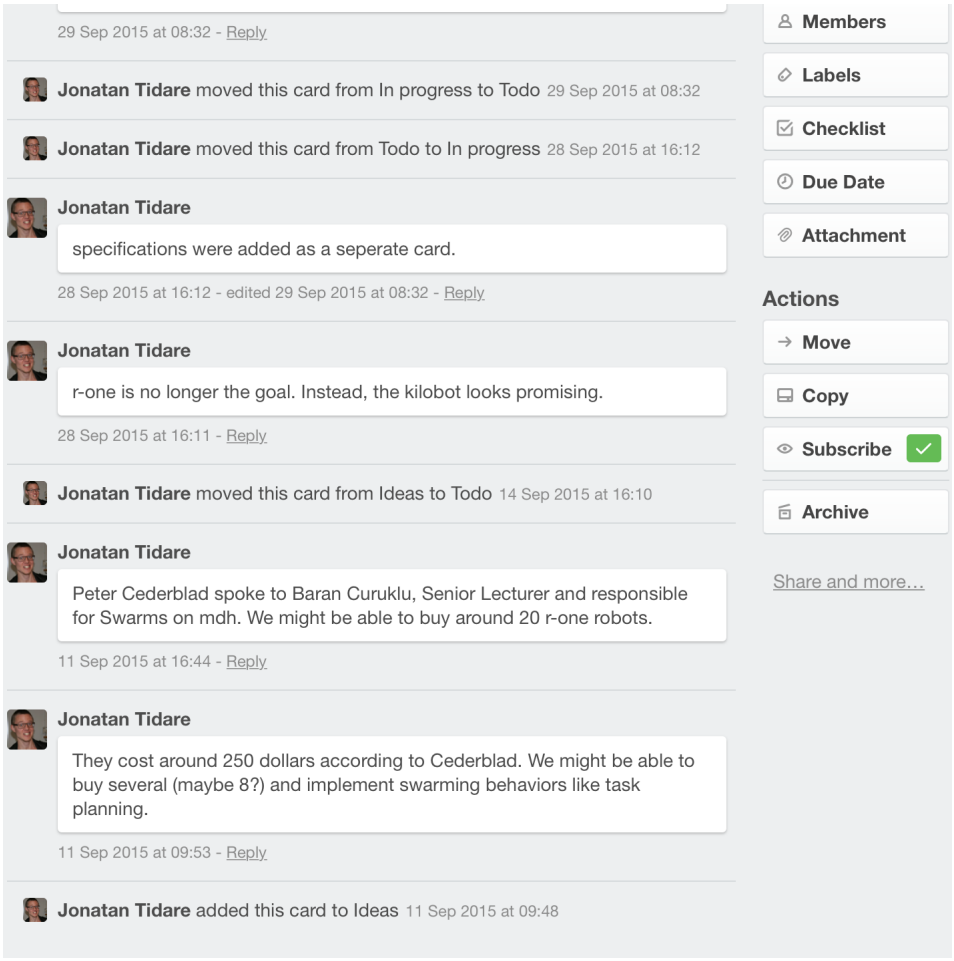
Move

Copy

Subscribe

Archive

Share and more…


29 Sep 2015 at 08:32 - Reply
Jonatan Tidare moved this card from In progress to Todo 29 Sep 2015 at 08:32
Jonatan Tidare moved this card from Todo to In progress 28 Sep 2015 at 16:12
Jonatan Tidare
specifications were added as a seperate card.
28 Sep 2015 at 16:12 - edited 29 Sep 2015 at 08:32 - Reply
Jonatan Tidare
r-one is no longer the goal. Instead, the kilobot looks promising.
28 Sep 2015 at 16:11 - Reply
Jonatan Tidare moved this card from Ideas to Todo 14 Sep 2015 at 16:10
Jonatan Tidare
Peter Cederblad spoke to Baran Curuklu, Senior Lecturer and responsible for Swarms on mdh. We might be able to buy around 20 r-one robots.
11 Sep 2015 at 16:44 - Reply
Jonatan Tidare
They cost around 250 dollars according to Cederblad. We might be able to buy several (maybe 8?) and implement swarming behaviors like task planning.
11 Sep 2015 at 09:53 - Reply
Jonatan Tidare added this card to Ideas 11 Sep 2015 at 09:48
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

Swarm Agent Spec in list In progress

Edit the description…

Attachments

Skärmklipp 2015-10-02 09.45.58.png
Added 2 Oct 2015 at 09:50
Open in New Tab · Remove Cover · Delete

Add an attachment…

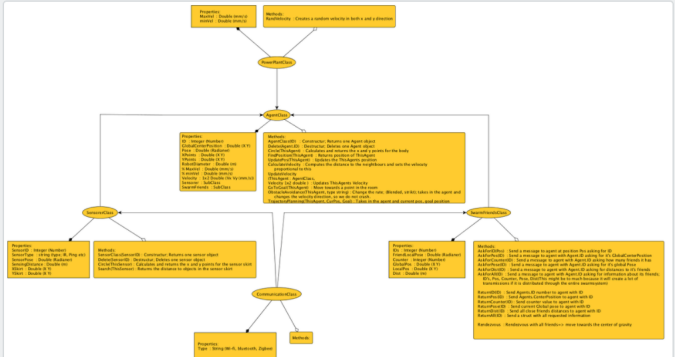
Add Comment

Write a comment…

Save Comment

Activity

Peter Cederblad attached Skärmklipp 2015-10-02 09.45.58.png to this card



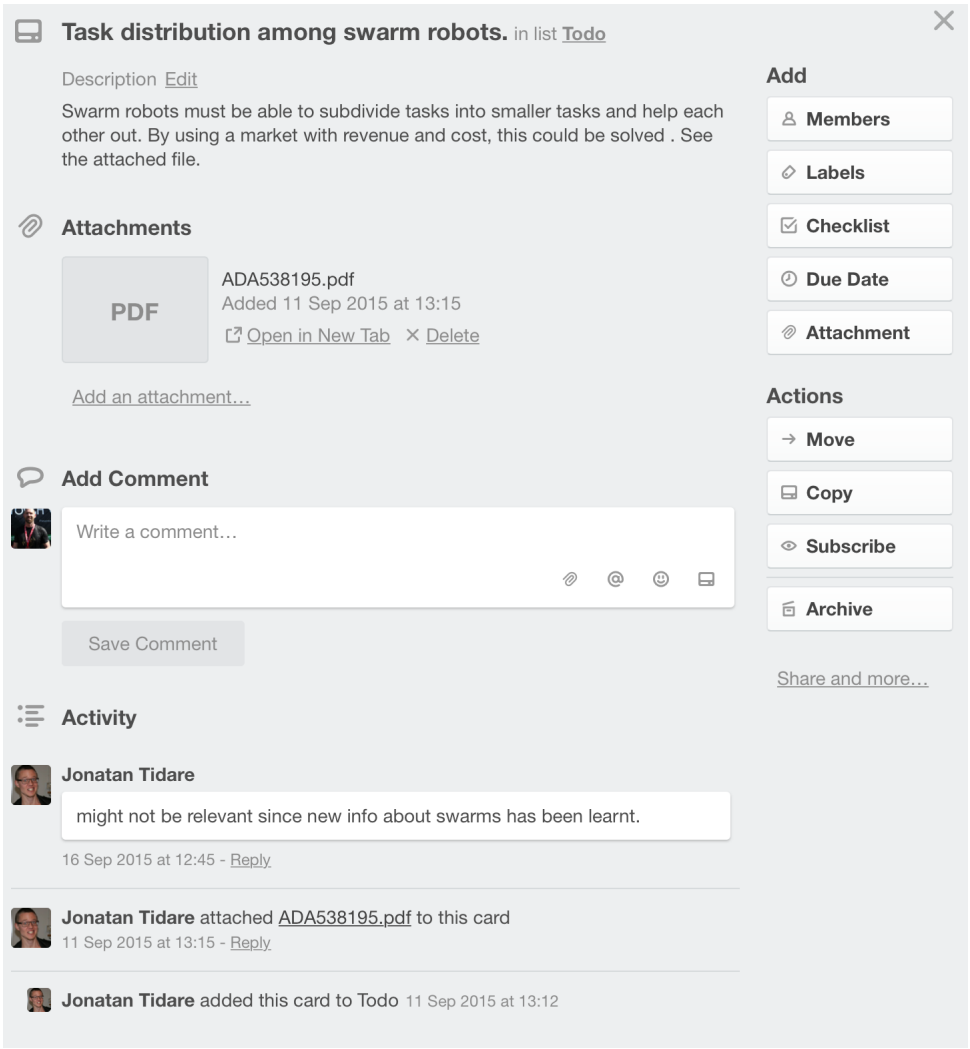
Add

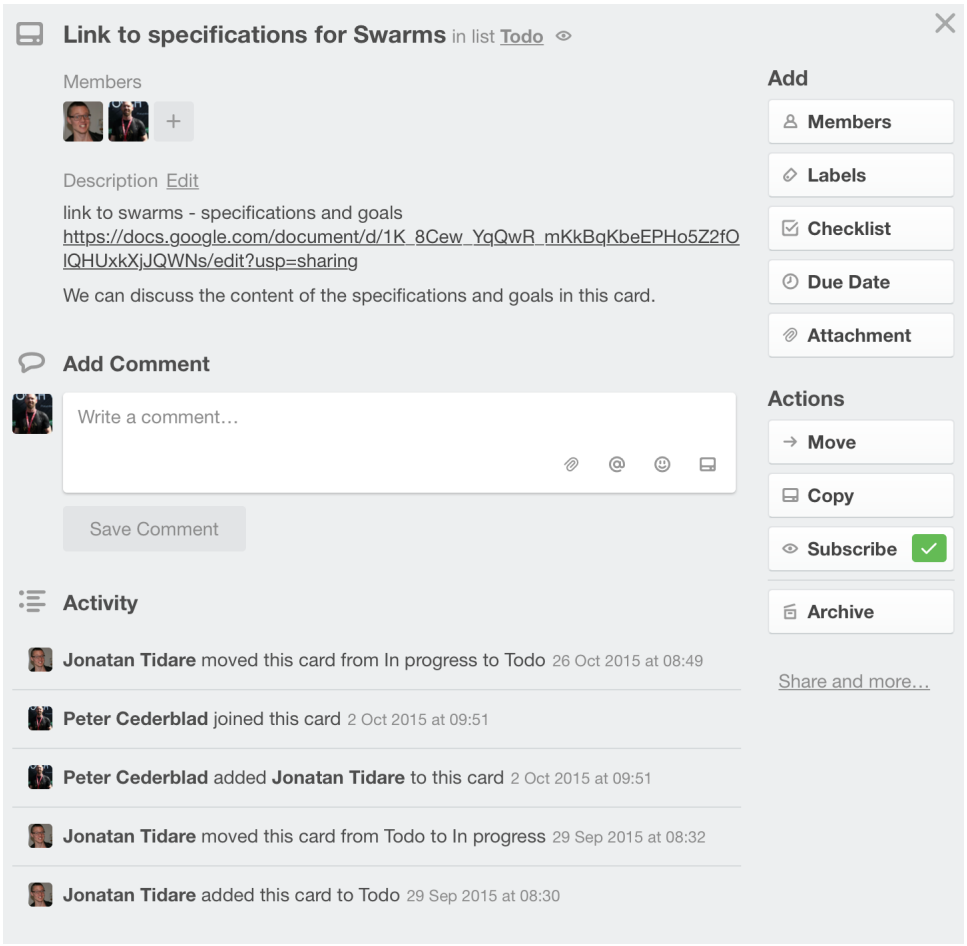
- Members
- Labels
- Checklist
- Due Date
- Attachment

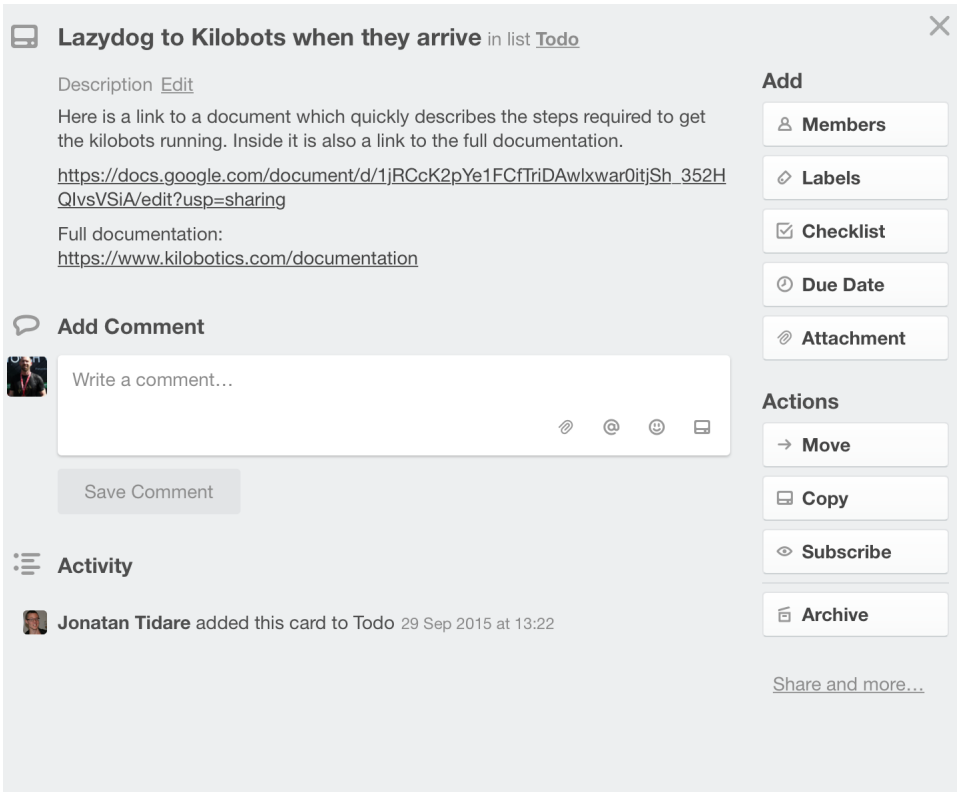
Actions

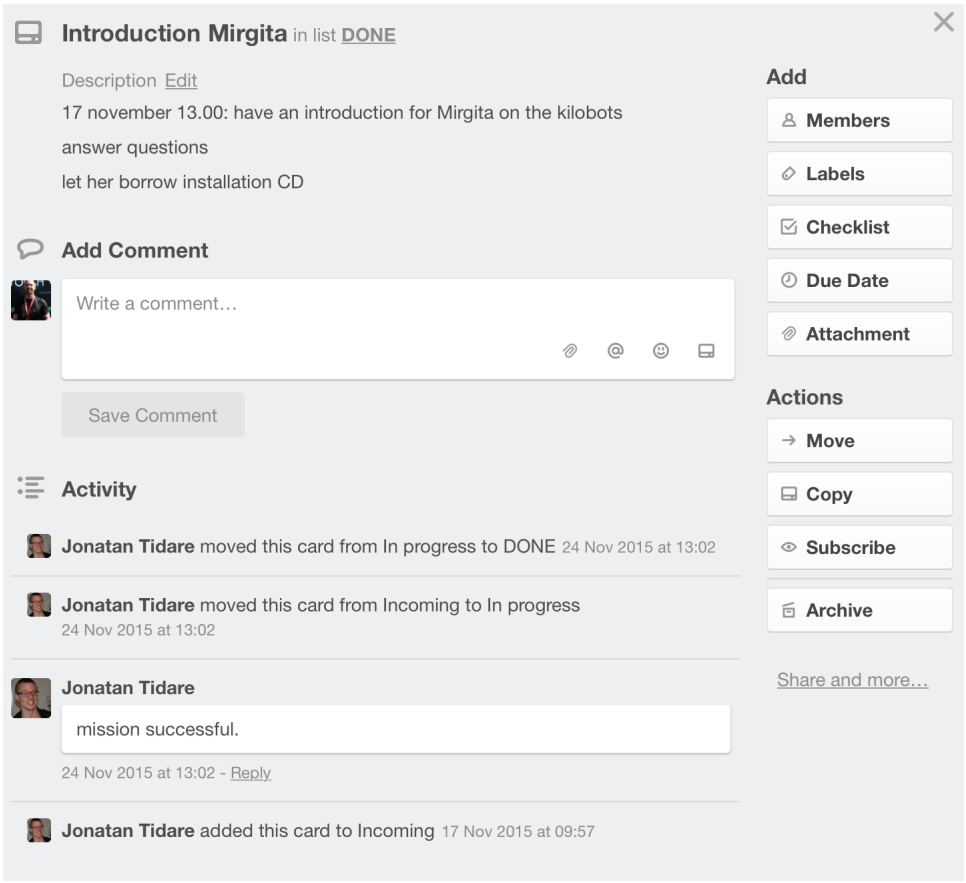
- Move
- Copy
- Subscribe
- Archive

Share and more…


Task distribution among swarm robots. in list Todo
Description Edit
Swarm robots must be able to subdivide tasks into smaller tasks and help each other out. By using a market with revenue and cost, this could be solved . See the attached file.
Attachments
PDF
ADA538195.pdf
Added 11 Sep 2015 at 13:15
Open in New Tab
Delete
Add an attachment...
Add Comment
Write a comment...
Save Comment
Activity
Jonatan Tidare
might not be relevant since new info about swarms has been learnt.
16 Sep 2015 at 12:45 - Reply
Jonatan Tidare attached ADA538195.pdf to this card
11 Sep 2015 at 13:15 - Reply
Jonatan Tidare added this card to Todo 11 Sep 2015 at 13:12
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more...


Link to specifications for Swarms in list Todo
Members
Description Edit
link to swarms - specifications and goals
https://docs.google.com/document/d/1K_8Cew_YqQwR_mKkBqKbeEPHo5Z2fOlQHUxkXjJQWNs/edit?usp=sharing
We can discuss the content of the specifications and goals in this card.
Add Comment
Write a comment…
Save Comment
Activity
Jonatan Tidare moved this card from In progress to Todo 26 Oct 2015 at 08:49
Peter Cederblad joined this card 2 Oct 2015 at 09:51
Peter Cederblad added Jonatan Tidare to this card 2 Oct 2015 at 09:51
Jonatan Tidare moved this card from Todo to In progress 29 Sep 2015 at 08:32
Jonatan Tidare added this card to Todo 29 Sep 2015 at 08:30
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…


Lazydog to Kilobots when they arrive in list Todo
Description Edit
Here is a link to a document which quickly describes the steps required to get the kilobots running. Inside it is also a link to the full documentation.
https://docs.google.com/document/d/1jRCcK2pYe1FCfTriDAwlxwar0itjSh_352HQIvsVSiA/edit?usp=sharing
Full documentation:
https://www.kilobotics.com/documentation
Add Comment
Write a comment…
Save Comment
Activity
Jonatan Tidare added this card to Todo 29 Sep 2015 at 13:22
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…


Introduction Mirgita in list DONE
Description Edit
17 november 13.00: have an introduction for Mirgita on the kilobots
answer questions
let her borrow installation CD
Add Comment
Write a comment…
Save Comment
Activity
Jonatan Tidare moved this card from In progress to DONE 24 Nov 2015 at 13:02
Jonatan Tidare moved this card from Incoming to In progress
24 Nov 2015 at 13:02
Jonatan Tidare
mission successful.
24 Nov 2015 at 13:02 - Reply
Jonatan Tidare added this card to Incoming 17 Nov 2015 at 09:57
Add
Members
Labels
Checklist
Due Date
Attachment
Actions
Move
Copy
Subscribe
Archive
Share and more…

F. Contact list

Name	Role	contact information
Daniel Adolfsson	Project leader-ROARy	dla.adolfsson@gmail.com 0702476546
Robin Andersson	Project leader-Butler	roband3333@hotmail.com 0708936629
Peter Cederblad	Project leader-Swarm	peter_cederblad@hotmail.com 0733320201
Joakim Kareliusson	Project leader-UAV	j.kareliusson@hotmail.com 0707653373
Ragnar Moberg	Mechanics pool leader	ragnar.moberg@gmail.com 0702375222
Oscar Svensson	Mechanics	hydralhammare@hotmail.com 0737826532
Albin Barklund	Hardware pool leader	albin.barklund@gmail.com 0707679968
Roxanne Anderberg	Hardware	roxanne.anderberg@hotmail.com 0762006256
Erick Vieyra	Hardware	sevieyra@gmail.com 0768255038
Emil Johansson	Hardware	emil.kex@gmail.com 0736812476
Jonatan Tidare	Software pool leader	jte11001@student.mdh.se 0704943558
Anders Olsson	Software	aon11013@student.mdh.se 0702487468
Mobin Hozhabri	Software	mhi14003@student.mdh.se 0737697724
Dennis Eklund	Software	eds04001@student.mdh.se 0764177124
Tobias Kriström	Software	tkristrom@gmail.com 0738272781
Rickard Holm	Software	holm.rickard@gmail.com 0732523205
Mattias Bäckström	Software	mattias.bax@live.se 0739050014
Ludvig Langborg	Software	lig11005@student.mdh.se 0730251316
Jakob Danielsson	Communication pool leader	jdn11003@student.mdh.se 0738761377
Nandinbaatar Tsog	Communication	ntg14001@student.mdh.se 0722290720
Tobias Andersson	Communication	tan10006@student.mdh.se 0735909074
Marcus Larsson	Communication	mln14013@student.mdh.se 0768000654
Sudhangathan Bankarusamy	Communication	sby14001@student.mdh.se 0768352404

References

[1] VectorNav. [Online]. Available:
https://ams.com/eng/content/download/302565/1085449

[2] VN-100 Rugged. [Online]. Available:
http://www.vectornav.com/products/vn100-rugged

[3] VN-100 User Manual. [Online]. Available:
http://www.vectornav.com/products/vn100-rugged/documentation

[4] MAX232AEWE+ datasheet. [Online]. Available:
http://www.farnell.com/datasheets/76791.pdf

[5] how to check max232 IC ok or damaged. [Online]. Available:
http://www.edaboard.com/thread223866.html

[6] RUBENSTEIN, Michael. Self-assembly and self-healing for robotic collectives. 2009. PhD Thesis. University of Southern California.

[7] Documentation on the original laborations. [Online.]Available:
https://www.kilobotics.com/labs

[8] Documentation on the kilobot platform. [Online.]Available:
http://ftp.k-team.com/kilobot/user_manual/Kilobot_UserManual.pdf