Swarming Module

Technical Document

Version 1.0

May 30, 2016

M. van Wilgenburg, W. Mukhtar, E. van Splunter, M. Siekerman, T. Zaal and M. Visser

List of Figures

List of Tables

Contents

1 Abstract

2 Introduction

This document describes the technical aspect of the "Swarming Module". This module is developed by students of the Amsterdam University of Applied sciences in collaboration with the Delft University of Technology. This project is a part of a running research program that is looking into the benefits of swarming compared to "standard" approaches, which would be one bigger and more complex robot doing all the work. Finally this program looks at the applications swarming might have on Mars.

This project contributes to the program by developing a so called "Swarming Module". This module will allow units in a swarm to determine the relative location to one another. When the units know their relative location, multiple complex tasks can be achieved like: path finding, payload transfer from one unit to another, autonomous recharging, and much more.

2.1 Swarming

The term "Swarming" is still a wide concept which is not well defined. However most people agree the following criteria should be met before a group of robots can be defined as a swarm:

- Autonomy - It is required that the individuals that make up the swarm-robotic system are autonomous robots. They are able to physically interact with the environment and affect it[1].
- Large number - A large number of units is required as well, so the cooperative behavior (and swarm intelligence) may occur. The minimum number is hard to define and justify. The swarm-robotic system can be made of few homogeneous groups of robots consisted of large number of units. Highly heterogeneous robot groups tend to fall outside swarm robotics.[1]
- Limited capabilities - The robots in a swarm should be relatively incapable or inefficien on their own with respect to the task at hand.[1]
- Scalability and robustness - A swarm-robotic system needs to be scalable and robust. Adding the new units will improve the performance of the overall system and on the other hand, loosing some units will not cause the catastrophic failure.[1]
- Distributed coordination - The robots in a swarm should only have local and limited sensing and communication abilities. The coordination between the robots is distributed. The use of a global channel for the coordination would influence the autonomy of the units.[1]

The swarming module provides the information so the last criteria 'Distributed coordination' can be met. The specifications must be formulated so they meet the criteria of "Large numbers", and "Scalability and robustness". It would be preferable to make the swarm endlessly scalable, however hardware limitations will probably hinder this.

2.2 Specifications

Before the features of the Swarming module are specified, swarming itself should be specified further. In section 1 of the research document this is done by creating a set of realistic scenarios. From this specifications like: minimal distance, maximal distance, maximum number of units, and precision are derived. These specifications are summarised in Table 1 and Table 2.

	Close range (0 - 3 m)	**Long range (3 - 25 m)**
Dead zone distance	0,015 meters	2 meters
Deviation (distance)	+/- 9%	+/- 1 meter
Angle Resolution	45%	-
Update frequency	40 Hz	1 Hz

Table 1: Swarming localization specifications

	minimum	**maximum**	**maximum (close range)**	**Preferred (demonstration)**
Number of Units	3	∞	130	6 to 12

Table 2: Specifications number of units

The choice is made to divide the the specifications for localization into two categories: Close range, and long range. This choice was made because at close range more precision precision is needed to prevent the robots from crashing into each other. While at long range the localization will be used not to loose the swarm. The maximum number at close range is derived from the theoretical number of units that can physically fit in a 3 meter radius. And the preferred number of units for demonstration purpose is used to set a goal for the project. For further clarification how these specifications are determined see chapter one of the research document.

3 Research

This section will a summary of the findings during the research phase. For the full research done during this project we refer to the ”Research Document”.
The research is split up in the following different subjects: Localization, Hardware communication protocol, Software communication protocol.

3.1 Localization

This section will cover the following subjects: Relative distance measurement, and Relative angle measurement. With these two variables the relative location of other units can be derived.

3.1.1 Relative distance

Before any calculations can be done to derive the angle of other units a distance must first be determined. The first choice that has to be made is what type of signal is used to do this. The most commonly used type of signal to measure distance are radio waves, which propagate with the speed of light. Hardware to measure these signals over short distances, would require clock-speeds of in the GHz, hence be expensive and unstable. Therefore the choice is made to use acoustic signals to measure the distance. This drastically lowers the requirements for the hardware that detects the signal.

Now that this is established there are still two ways to implement a distance measurement. These are: Time of flight (TOF), and Received signal strenght (RSSI). During experiments (found in section 4.3 of the Research document) we found that the received amplitude of the acoustic signal was not proportional to the distance. We also found that the signal was still well defined over the specified distance of three meters. Because of this the signal can be detected and used for TOF.

A system is proposed that combines the two signals (radio and acoustic) in one system. Swarming modules each get their place in time to send their acoustic signal. At the moment the module starts sending the acoustic signal, he sends a message over the radio communication channel saying "im going to send my signal", the radio signal arrives close to instantly compared to the acoustic signal. At this point all the other modules start their timers, and stop them when the acoustic signal arrives a few moments later. The distance can now be calculated to multiply the time with the speed of sound.

3.2 Relative angle

Determining the relative orientation with respect to each other can be done in various ways. some methods involve larger limitations than others. In general angular measurements are done using goniometric equations. It was found that calculating the angle with a single measuring point on every swarming module has one big problem. Because geometric functions are used to calculate the angle, there will always be two solutions to the equation (see figure 1 and section 3.1.1 of the Research document). To solve this the units would have to move and recalculate again to get the right answer. This would limit the units in their movement and functionality and is non desirable.

$$Because\ Xp = Xq,\ cos(-\alpha) = cos(\alpha)\ applies \tag{1}$$

$$Because\ Yq = -Yp,\ sin(-\alpha) = -sin(\alpha)\ applies \tag{2}$$

To solve this problem a system is proposed with three or more acoustic receivers with a predefined distance between them. Using the difference in time and the predefined distance, the angle can be calculated with only one solution. This method is shown in figure 2.

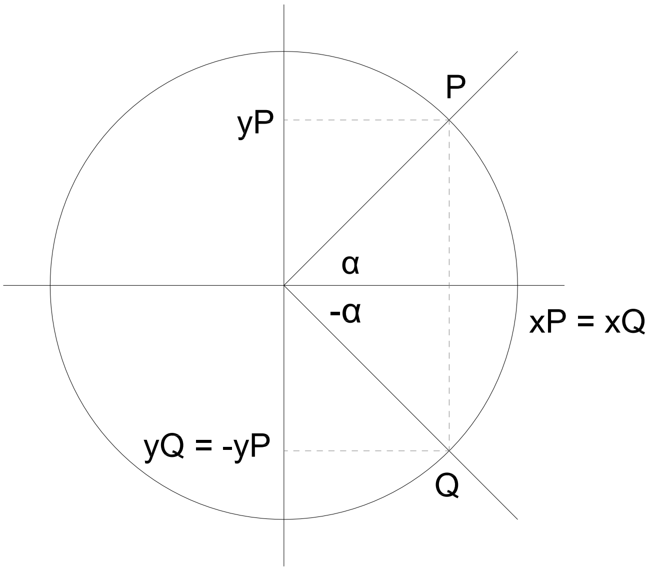

Figure 1: Unit circle where points P and Q are mirrored on the X-axes

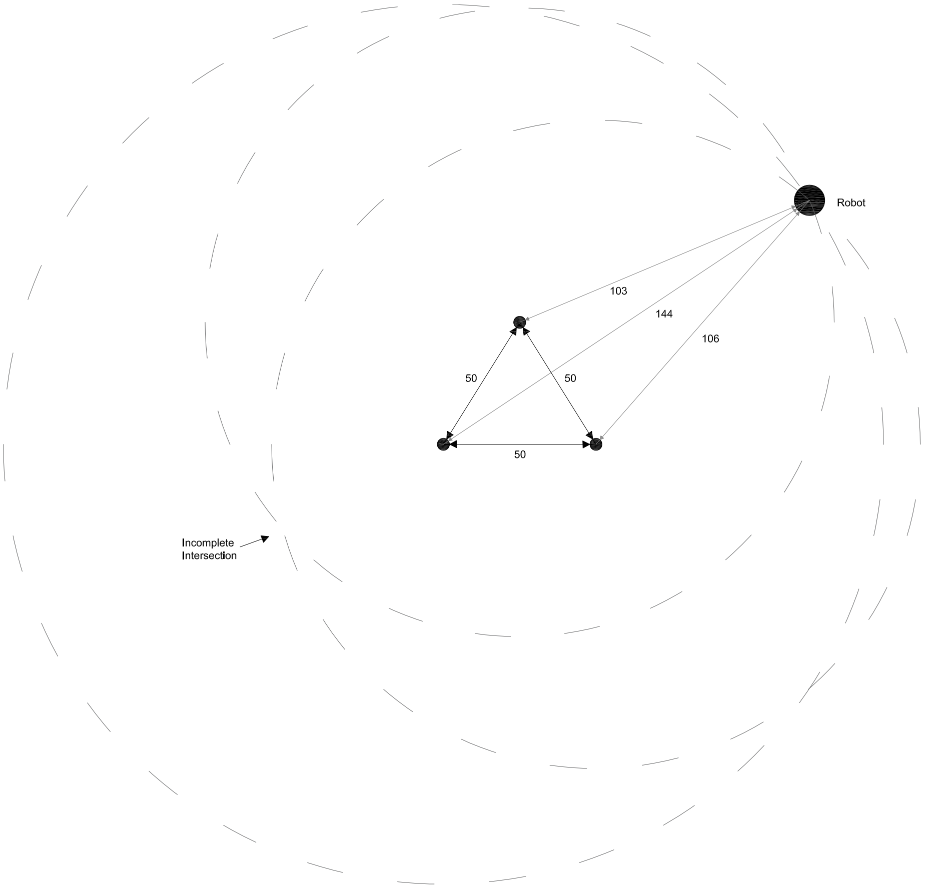

Figure 2: Angle determination using trigonometry

3.3 Demodulation methods

As talked about before the signal will be modulated and demodulated to introduce a reference point in the signal. This reference point is used to determine the exact time the signal has travelled. Also the incoming signal from the microphone will suffer from noise and interference. Because of this the demodulator should be noise resistant. Also its part of the specification of the swarming module that it should be a robust system. Therefore the use of high clock micro controllers and complicated software are not preferred.

Project members had experience with frequency demodulation using an analog PLL, and were confident this could be easily tested relatively quick. This will be discussed in the next subsection.

3.3.1 Analog Phase locked loop

As discussed in section 9.3 of the technical design a PLL is used to demodulate the signal. A window comparator translates the VCO-in voltage into binary code. To test the precision in time a test set-up was build. Two xmega128a4u were used, one for the demodulator side and one for de modulator side see figure

3 . During tests the PLL was able to lock on the acoustic signal from a maximum distance of 4 meters which is well within the specification of 3 meters. The PLL was also able to demodulate the FSK signal at this range. So now the reference point could be identified by the demodulator.

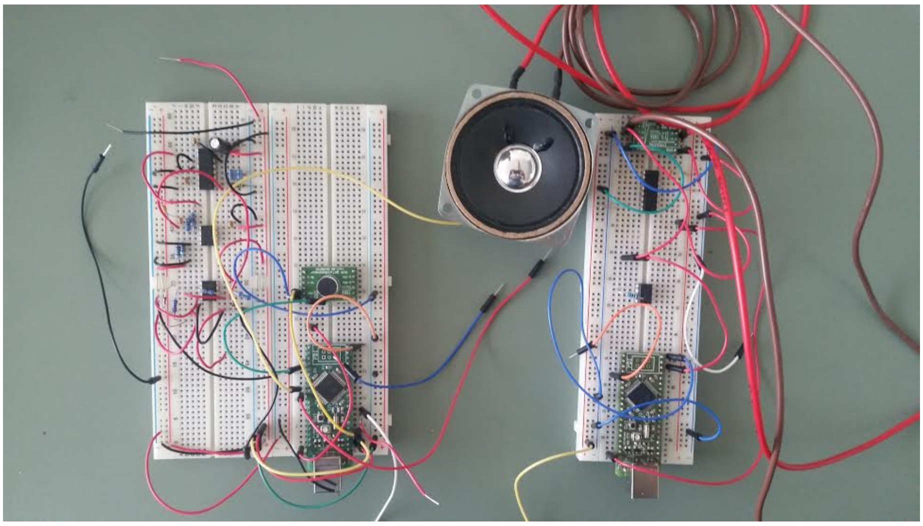

Figure 3: Test set-up, the right breadboard is de modulator with a speaker, the left breadboard is the demodulator with an microphone and the PLL

Measure point	1	2	3	4	5	6	7	8	9	10
0,1m	3115769	3011495	3048980	3048133	3113623	3049715	3112041	3114898	3041495	3115769
1m	3112100	3012334	3102004	3159832	3068347	3123548	3110398	3119478	3062395	3119325
3m	3130596	3105236	3120056	3125069	3140686	3101498	3130295	3049821	3101956	3120549

Table 3: Measured times in clock pulses, pre scaler is set to 1024

The modulator and the demodulator are connected with a wire. The modulator sends a bit when it starts to send the acoustic signal. This bit interrupts the demodulator which starts a timer. When the reference point in the acoustic signal is detected by the PLL the windows comparator sends a binary ’1’ to the micro-controller which stops the timer. The timer values were then printed in a terminal and are documented in table 3 The measured times are printed in clock pulses with a pre-scaler of 1024. At close range the measured times seem the most stable. However some big deviations of around 500.000 clock pulses still exist. When these times are used to calculate the distance this would mean a deviation is distance of around 5 meters, which is unacceptable. The digital part of the demodulator is tested by connecting a frequency generator to it and observing the time deviation. The time deviation was at a maximum of 200 clock pulses, which is acceptable. The timing deviation is clearly created by the PLL. In section 9.3 of the research document was stated that the PLL would introduce a constant time deviation because of the RC time introduced by the loop filter. We suspect that the reason for the deviation is caused by the window comparator. The time the windows comparator takes to switch from 0 to 5 volt might deviate, which in turn causes the deviation in time. We have not found a way to test or measure this assumption though. Because if this problem the analog PLL approach is not usable for this project.

3.3.2 Demodulating with a Microproccessor

Its been established that the analog solution does not meet the criteria. The other option is to process the signal digitally, with an microprocessor. The ADC of the Xmega contain an comparator. This can be used to detect how many times the incoming signal goes trough a certain threshold. From this information the frequency can be derived.

3.4 Acoustic Sensing

In this section the behaviour of acoustic signals will be researched. This was done in real life experiments using a microphone, speaker and scope to analyse the received signals. During experiments the following points where analysed:

- Sound has to spread in every direction (omnidirectional).
- The received acoustic signal has to be well defined within a minimal distance of 3 meters.
- What frequency shows the best results.

More details about the experiment and the setup are shown in section 4 of the Research document. The first experiment had the speakers facing the microphone directly, this was clearly the best situation, higher frequencies showed higher amplitude and the signal was well defined. However the most ideal situation would be for one speaker to be omnidirectional instead of having to use multiple speakers.
Another test was done with the speaker and microphone both facing upwards. It was observed that higher frequency sound (around 5 kHz) becomes more directional. The best results were achieved around 3 - 4 kHz, this was the point where the highest amplitude was achieved and the minimum distance of three meters was easily met.

3.5 Functional specifications

3.5.1 Wireless communication

The specifications of the wireless communication are discussed in our framework, but will be shortly summarized.

Table 4: Wireless communication module

Module	Wireless communication module
Input	Communication with the system Communication with the other robots in the swarm
Output	Communication with the system Communication with the other robots in the swarm
Functions	establishes the communication between the modules swarm Obtains information regarding the distance between the different robots

To communicate between different units a communication network should be set up with a common protocol . In addition, the protocol must be able to provide additional information such as localization. See Table 4.

3.5.2 Central control unit

The central control unit handles all communication between the various modules that are present on the robot. See Table 5

Table 5: Central control unit

Module	Central control unit
Input	All information from the system
Output	All control of the modules in the robot
Functions	All the communication comes together and controls the inputs and outputs

3.6 Specifications

3.6.1 Wireless communication

In Table 6 a more detailed version of the specifications are written down. From these specifications the module will be implemented as is shown in the next section.

Table 6: Wireless communication module

Module	Wireless communication module
Input	5 VDC Swarm data RF signal
Outputs	RF signal Swarm data
Function	Processes information to and from other swarm modules. Transmission speed is not defined yet. The transmission range must be around 100 meters or more.

3.6.2 Central operating unit

The central operating unit processes data from the sensors and swarm network. This unit uses the external wireless swarm communication, the local communication bus and the debug interface as an communication channel. The local communication bus uses a TWI (Two Wire Interface) protocol and is used for the communication with the robots peripherals i.e. actuators. The debug interface of the central operating unit uses an UART protocol to communicate with external systems for debug purposes.

Table 8: Central operating unit

Parameter	**Value**
Flash (kBytes):	128 kBytes
Pin Count:	44
Max. Operating Freq. (MHz):	32 MHz
CPU:	8-bit AVR
Max I/O pins:	34
USB Interface:	Device
SPI:	7
TWI (I2C):	2
UART:	5
ADC Channels:	12
ADC Resolution (bits):	12
ADC Speed (ksps):	2000
DAC Channels:	2
SRAM (kBytes):	8
EEPROM (Bytes)	2048
Operating Voltage (Vcc):	1.6 to 3.6

Table 7: Central operating unit

Module	Central operating unit
Input	3,3 VDC Data streams (TWI, UART)
Outputs	Data streams (TWI, UART)
Function	Calculates the relative position to other swarming modules. Distributes local and external data streams. Analyses and uses sensor information.

3.7 Implementation

3.7.1 Central operating unit

In Table 7 the basic specifications of the AtXmega128A4U are shown. This is the microcontroller used in as the central operating unit. The microcontroller from Atmel complies with all the specifications. There are also newer micro controllers available, but are not yet implemented for this project, since the older versions can handle the workload and work fine. There are two I^2C connections available on the microcontroller which are needed to connect the module to the robot. Also the multiple ADC and DAC connections can come in handy if any sensors or actuators need to be added to the module. Overall the AtXmega128A4U fits all the specification that are required, and there is room to expand the functionality when it is necessary. An overview of all the specification can be found on the website of Atmel.

The Central operating unit handles multiple functions to make the group of robots a swarm. Those functions will be explained in this section. There are four major functions that are implemented in the microcontroller. Controlling communication that goes in and out, storage and maintain data, calculating the population of the swarm and send and receive message that are used for localization. The communication will be mainly about the communication with the Swarmbee module. The connection to the robot is just as important but will not be thoroughly discussed in this document. First of all we will discuss the implementation of the communication.

3.7.2 Communication

The central operating unit needs to send request and receive data from the Swarmbee module. This information is transmitted using the UART protocol. The communication is mainly used for sending API commands to the Swarmbee module. The UART connecting uses a baudrate of 115200 bps since this is the transmission speed of the Swarmbee module. The Xmega uses two pins, RX (PC2), TX PC3) to achieve a wired connection with the Swarmbee. The module needs to be able to communicate with a robot platform. To communicate with this platform the module uses I^2C communication. This should be standard for every robot, so that the module can easily be placed on any type of robot platform. For debugging pin C7 en C8 are reserved. The debugging connecting can be established with a terminal such as Putty.

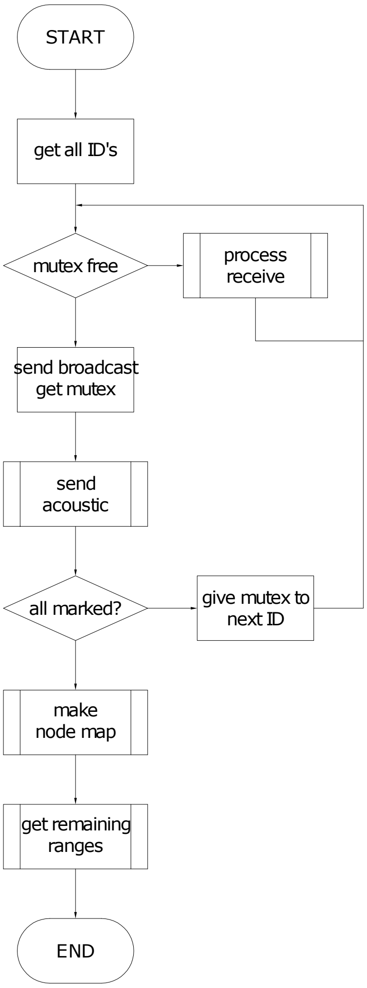

Figure 4: main algorithm for updating the swarm population including range requests

3.7.3 Swarm population

To update the swarm population all modules need to receive all node ID's from other modules in range of the Swarmbee module. This is the first step in the flowchart of the main algorithm shown in figure 4. The swarm bee automatically assigns a unique ID to all nodes. A list of all nodes and the corresponding ranges can be requested from the swarm by a API broadcast message: "GRWL", or, "get ranging white list". This broadcast message can be requested at a certain interval which depends on the demand of this information. When the swarm bee receives a ranging request its response will look somewhat like this:

```
*RRN:1F3123123133,1F3CFF322133,0,001843,04,-56
```

Where the first two hexadecimal values represent the source ID (sending module), and the destination ID (receiving module). This is followed up by an error code and the range relative in centimetres to the destination node. The end of the message contains a custom value which can be set by a ”notification configuration”-request. For example, this value can represent different sensor values like; x,y,z acceleration, the measured RSSI value or the current battery level.

3.7.4 Acoustic sensing send/ receive

Using the information obtained by this message, the population list can be filled with ID’s. after this is done the first swarm unit in queue will get the mutual exclusion to be the only sender in the acoustic ranging process, it will enter the ”send acoustic”-process, figure 5.1. since only one member can be the sending module, it will broadcast that the mutex is taken, now the rest of the swarm will enter the ”process receive” algorithm, figure 5.2. When the swarm module is in sending mode it will send out a sound signal and retrieve the ranging results form the receiving modules one by one. The receiving modules will start the localization algorithm when a request from the sender is received. When there is no result a node will be marked as ”out of range”. when a sender retrieved all ranges this node gets marked as ’done’. The main algorithm, figure 4, will now give the mutex to the next swarm member until all units are marked. When all nodes are marked, each module will create its own node map. when there are any missing ranges these gap will be filled by the ranges received by the swarm bee in the ranging request broadcast message.

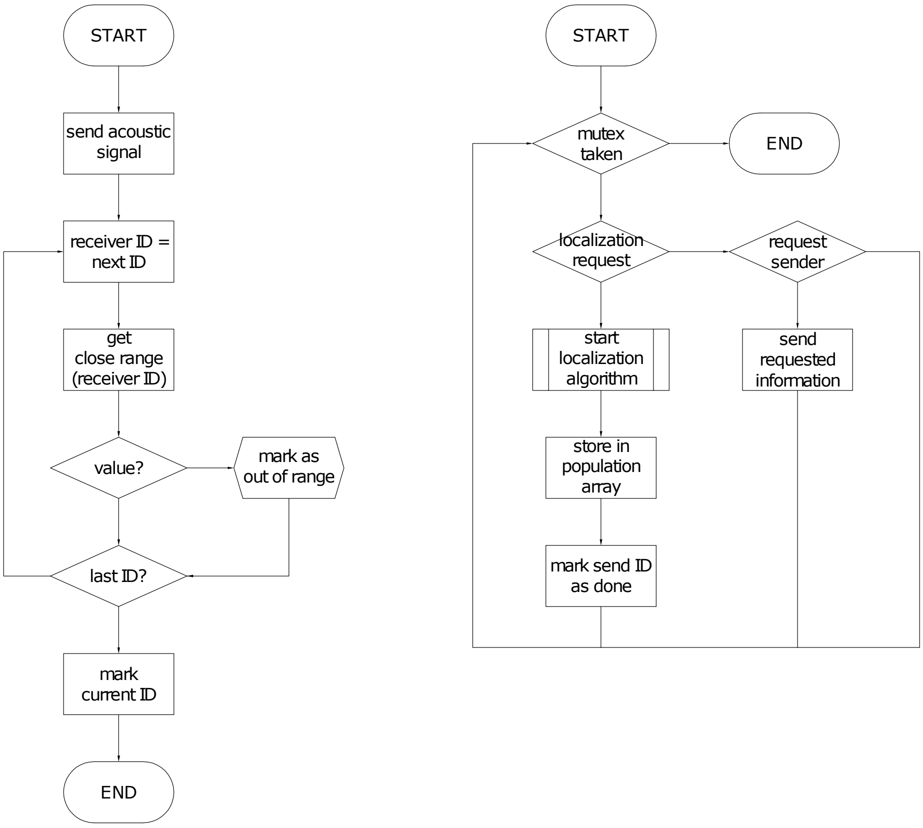

Figure 5: 1:"send acoustic" when the swarm module is in sensing state 2:"process receive", when the swarm module is in receiving state

3.7.5 Wireless communication

For the wireless communication module the most important function is that it can transmit information over atleast 100 meters. Since the specification states that the swarm must be detected within this range. The speed of de communication is not defined yet, but needs to be fast enough to handle all the messages. The Swarmbee LE module will be used for the wireless communication. Its uses the ISM band, 2.4 GHz to transmit and receive the data. The maximum range of the module relies on the environment in which it is used. Under ideal conditions the Swarmbee can reach a distance of 1200 meters. It depends on how many obstacles, reflections and interference there is to disturb the signal. An experiment from Nanotron showed that until a range of 150 meters the ranging success rate is about 100% with a transmission speed of 155 kbps. Experiments we have done ourself show that the antenna on the device does not respond very well inside a building. Test outdoors should still be done, but there is a possibility that an external antenna is needed to reach these distances. The actual speed is also not tested yet, but could be done in the future. Transmitting data between the swarm modules can be configured with a speed of 250 Kbps or 1 Mbps. Experience shows that these speeds are more then enough to fit the needs. This module fits all the specifications and is suitable for the use of

Table 9: Swarmbee LE module

Parameter	**Value**
Frequency range	ISM band 2.4 GHz (2.4 - 2.4835 GHz)
Modulation	Chirp Spread Spectrum (CSS)
Transmission modes	80 MHz, 1 Mbps or 250 Kbps (80/1 or 80/4 mode)
TOA capture accuracy	<1 ns (better than 30 cm)
Typical air time per ranging cycle	1.8 ms
RF output power	configurable - 22 t0 16 dBm
RF sensitivity	-89 dBm typ. @80/1 mode -95 dBm typ. @80/4 mode
RF interface	50 Ohm RF port (for external antenna)
Host interface (UART)	500bps ∼ 2 Mbps
Power supply	3 - 5.5 V
Max. supply voltage ripple	20 mVpp
Active power consumption*	120 mA during transmission, 60 mA during receive in 80/1 mode
Power consumption in sleep mode*	5.5 mA (transceiver disabled, all peripherals on)
Power consumption in snooze mode*	4.5 microA (transceiver disabled, all peripherals off, wake-up by timer)
Power consumption in nap mode**	4.5 ∼ 600 microA (transceiver disabled, all peripherals off, wake-up by interrupt)
Power consumption in deep-sleep mode*	<1 microA (device completely disabled)
Operating temperature range	-30 - 85 °C
Dimensions	40 mm x 24 mm x 3.5 mm
Weight	7 g
*Power consumption in all modes is measured at 20°C, 3.3 V.	
**Power consumption in nap mode depends on interrupt sources (GPIO pins or MEMS or both).	

wireless communication.

4 Functional specification

This chapter describes the functional design of the swarming module. An overview of the swarming module is given in figure 6. The module is designed in such a way that it is modular. This module can be placed on an object (i.e. a robot) and can be used by the object to send and receive relevant information about other in-range swarming modules.

This chapter discusses the following subjects:

- Swarming module
- Power supply and safety
- Central operating unit
- Wireless communication module
- Relative orientation sensor
- Orientation sensor
- Localisation sensor

4.1 Swarming module

The overall design of the swarming module is shown in figure 6. Each block of the overall design is described in the following figures. The central operating unit is at the hearth off the swarming module, and takes care of the localization algorithm.

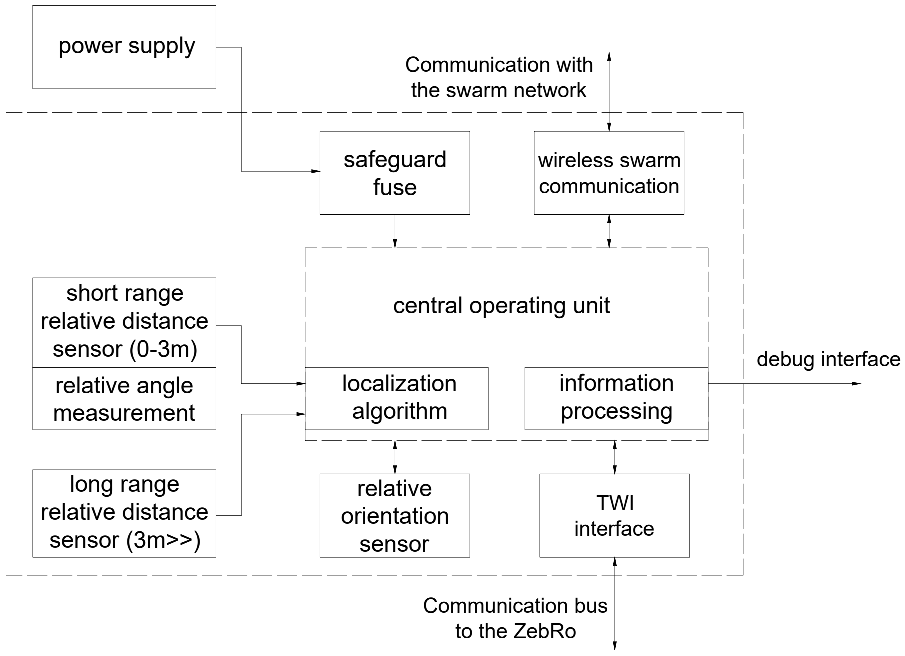
power supply
Communication with the swarm network
safeguard fuse
wireless swarm communication
central operating unit
short range relative distance sensor (0-3m)
relative angle measurement
localization algorithm
information processing
debug interface
long range relative distance sensor (3m>>)
relative orientation sensor
TWI interface
Communication bus to the ZebRo

Figure 6: Overview of the swarm module design

4.2 Short range localisation sensor

The principle of the short range localisation sensor is displayed in figure 8 and 7. As mentioned earlier the short relative distance measurement will be done acoustically. With acoustic sensing it is required to let the speaker sway to the right oscillation frequency, reaching this oscillation frequency takes time since the speaker won't oscillate at the right frequency instantly. This complicates the measurement a little bit, since just detecting the to be measured signal won't always translate into an exact distance due to the time the speakers take to sway into oscillation.

A solution to this problem is to create a fixed reference point within the sent signal. Creating this reference point after a fixed time enables the speaker to sway into oscillation making sure that no faulty measurements occur do to this "swaying time". When the reference point has been detected by the receiver an accurate distance can be determined by subtracting the fixed "sway time".

Without the implementation of this fixed reference point it is possible for a receiver to miss a few periods of the signal, due to the speaker still swaying into oscillation and the time of swaying being unknown. For example take an acoustic signal (Speed of sound 340.29 m/s) with a frequency of 4000Hz, one period of this signal is 0.00025s. Missing one period of this signal translates into a distance error of 8.5cm. Since the sway time of a speaker can change over time due to mechanical stresses a fixed time stamp seems the most suitable solution for the long term. The best way to create this reference point it still to be determined. This will be done by either frequency shift keying or phase shift modulation. Some real like testing will have to determine the best method.

4.2.1 Modulation

The modulation method is going to be determined by how well the reference point can be recognized by the demodulator. In the research phase, test were done with an speaker and a microphone. This showed that frequency and phase were well defined over distance but amplitude was not. Because of this the available methods for modulation of the acoustic signal are frequency modulation or phase modulation. The most important feature of the modulator must be consistency, meaning that the reference point should be send at exactly the same point in the signal every time. This is so that no deviation will occur when compensating for the missed periods before the reference point. In theory this can be done by both of the methods.

See figure 7 for a schematic representation of the modulator.

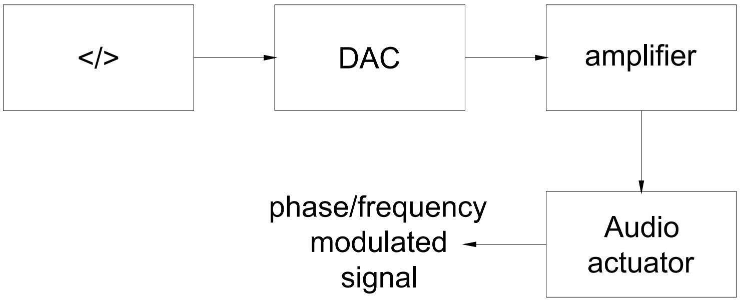

Figure 7: Modulator

4.2.2 Demodulation

A demodulator will be needed to retrieve the digital signal send from one swarm-module to another. One swarming module will send an acoustic signal to the other modules, this signal will then be picked up by an sensor and will need to be processed properly before it can be used to determine the time it took the acoustic signal to travel from one swarming-module to the other. The signal retrieved by the acoustic sensor will also pick up a lot of noise and other sounds from the environment. Also, when further away the signal might be low in amplitude. To increase the amplitude and lower the noise, the signal will be amplified and filtered. These modifications to the signal will improve the signal to noise ratio drastically but a lot of noise might still remain. Therefore the demodulator itself must be insensitive to noise. The main purpose is the demodulator is to identify the reference point. And more importantly, determine the point in time the reference point is spotted. The time determined will be used to calculate the distance. Hence, the precision in which the demodulator determines the reference point in time effects the precision of the distance measurement. Only one swarming module will be sending his acoustic signal at any moment. When there are 10 units in its specified range and a preferred update frequency of 40Hz (see section 1). This gives every swarming-module only a small window in time to send their signal and for the other to receive it. When a frequency of 4kHz is chosen for the signal, every swarming-module will have 10 periods to send their signal. There for the demodulator must be able to lock onto the signal and determine the reference point within a few periods of the acoustic signal. See figure 8 for a schematic representation of the demodulator.

Table 10: Demodulation specifications

Module	Demodulator
Input	FSK Signal
Outputs	Reference point
Function	Demodulation, Reference point detection
Features	Noise insensitive, precision (time), Quick demodulation

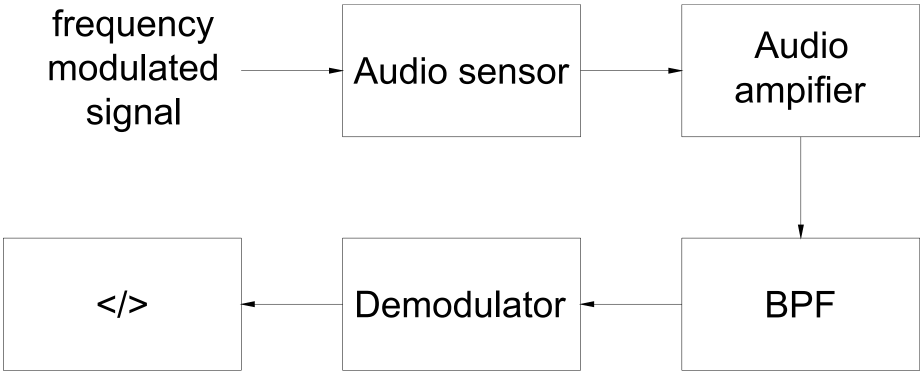

Figure 8: Demodulator

5 Implementation

Now that specifications of the functional blocks are defined, a implementation for these blocks will be found, using the information from the research phase. Just like section Functional specifications every block will be discussed independently. How all of the functional blocks fit together will be discussed in the section for the central processing unit.

5.1 Short range localization sensor

The short range localization consist of multiple smaller parts: Acoustic actuator/sensor, modulation/demodulation, distance measurement, and angle algorithm.

5.1.1 Acoustic actuator sensor

As stated earlier a short range localization will be implemented using acoustic wave signals. The sensor will be implemented using microphones and an actuator will be implemented using a speaker.

For Trigonometry angle determination the module must atleast poses three microphones spread in a triangle where all sides have the same distance as shown in figure 9.

For the distance between the different microphones a length of 10 cm is chosen because with the velocity propagation of sound being 343.2 m/s (at 20°C Celsius) and the underlying distance between de microphones being 10 cm using an 32 MHz micro-controller should grant enough overhead time to process the three different times being detected by the microphones.

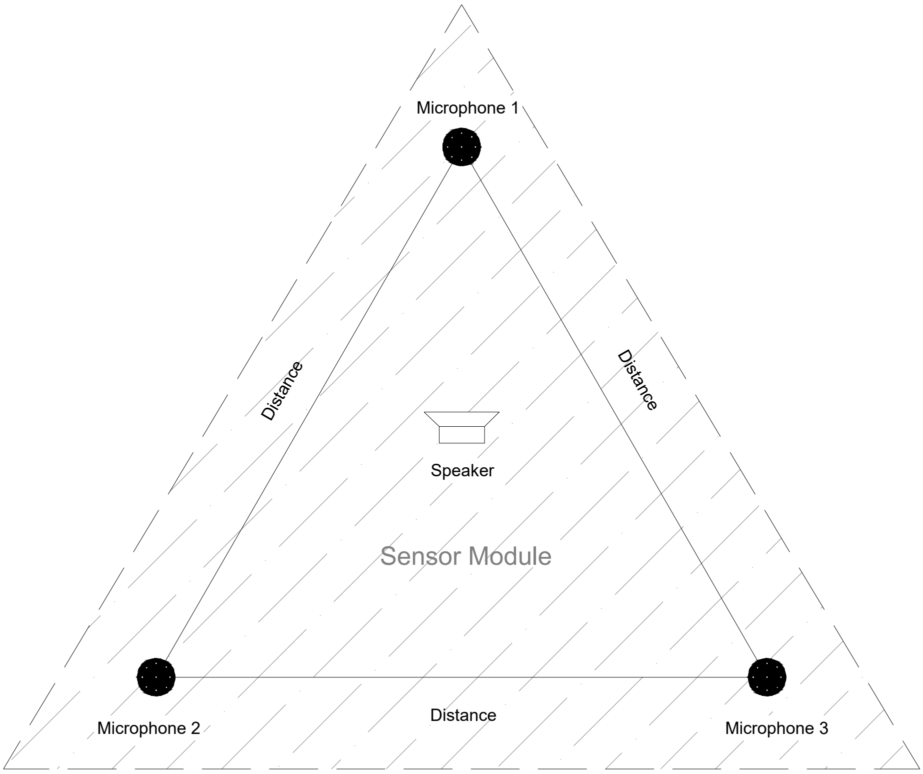

Figure 9: Localization Module

The acoustic signal is send by a speaker facing upward. A metal plate is placed a 3 mm above the speaker, tests during the research phase showed this greatly improved the omnidirectional spread of the signal. The signal that's send it chosen to have a frequency of 4 kHz which has the best omnidirectional and range properties (section 4 of the research document).

The signal received by the microphone, translates in just a few millivolts. This signal is amplified for a total of 9600 by several stages of amplifiers, so that the signal clips on the supply voltage. This ruins the shape of the signal. However we're only interested in the frequency which will remain the same. After that the signal is bandpass filtered, to get rid of most of the noise. After this the signal is fed into the ADC of the Xmega128a4u. The modulator and demodulator will now be discussed in detail.

Modulator
The DAC of the Xmega128a4u is used to create a 4 kHz square wave signal. The signal will be send out for a pre-defined period of time and then the signal will pause for one period and continue. The pause will be the reference signal. The reference point is to eliminate certain uncertainties like the time it takes for the speaker to start resonating. The signal send looks like this 10 periods of signal, 2 periods nothing 10 periods of signal.
ADD block diagram of modulator

Demodulator The demodulator has to be able to detect the frequency of the

incoming signal. The demodulator has to differentiate between the wanted signal and other signal coming from the environment. This is done by looking at the characteristics of the signal, most importantly the frequency. The comparator in the ADC of the Xmega is used to count how many times the signal crosses a certain threshold. The frequency is derived from the simple fact that two crossings, should mean one period has passed. The reference-point is found when the micro-controller detect the send-pause-send sequence that's send by the modulator. Important to note is that the micro-controller looks at the time of the pause moment to make sure its the actual signal.
ADD block diagram of demodulator

5.1.2 Distance measurement

The distance is derived from the time it takes the acoustic signal to travel from module A to module B. Suppose that module A is about to send its signal. Just before it starts sending the acoustic signal, it sends a message over the radio communication, which is almost instant compared to the speed of the sound waves. Module B starts its timer and wait for the signal to arrive. When it arrives it stops it's timer and takes off some pre-defined corrections based on tests done earlier. The distance can now be derived multiplying the speed of sound with the measured time. **ADD state-diagram distance measurement**

5.1.3 Angle measurement

The Swarming module has three microphones placed with known distances between them. The relative angle is derived from the time measured with each microphone, and the distance between the microphones. In section 3.2 there's more information on how this is done.

The following algorithm shows pseudo-code how the angle will be calculated.

```
ISR(reference call){
  start timer;
}
ISR(mic 1){
  Time 1:= Timer;
  recieved:=recieved+1;
}
ISR(mic 2){
  Time 2:= Timer;
  recieved:=recieved+1;
}
ISR(mic 3){
  Time 3:= Timer;
  recieved:=recieved+1;
}

While recieved = 3{
  time_1:= time_1-overhead_time;  //remove overhead time
  time_2:= time_2-overhead_time;  //remove overhead time
  time_3:= time_3-overhead_time;  //remove overhead time
  length_1:=time_1*speed_of_sound/clock_freq
  length_2:=time_2*speed_of_sound/clock_freq
  length_3:=time_3*speed_of_sound/clock_freq
  angle_1:=acos((length_1^2+mic_distance^2-length_2^2)/(2*length_1*mic_distance));
  angle_2:=acos((length_2^2+mic_distance^2-length_3^2)/(2*length_2*mic_distance));
  angle_3:=acos((length_3^2+mic_distance^2-length_1^2)/(2*length_3*mic_distance));
  read compas_angle;
  angle_1:=angle_1+compas_angle;
  angle_2:=angle_2+compas_angle;
  angle_3:=angle_3+compas_angle;
  x_1:=sin(angle_1)*length_1;
  x_2:=sin(angle_2)*length_2;
  x_3:=sin(angle_3)*length_3;
  y_1:=cos(angle_1)*length_1;
  y_2:=cos(angle_2)*length_2;
  y_3:=cos(angle_3)*length_3;
  x_center=x_1+center_distance_x;
  y_center:=Y_1+center_distance_y;
  recieved:=0;
}
```

Code 1: Algorithm to detect the location

6 Bibliografie

References

[1] *Swarm Intelligence and Its Applications in Swarm Robotics*, D.A. Aleksandar Jevtic, 1 2015, vertrouwlijk.