Software

Table of content

[Block diagramm 2](#_Toc496995830)

[List of robots 3](#_Toc496995831)

[Data of the own robot 3](#_Toc496995832)

[Data of the other Robots 4](#_Toc496995833)

[Pathfinding with A\* 5](#_Toc496995834)

[Determining value of status LED 6](#_Toc496995835)

[Flow cart 7](#_Toc496995836)

[Protocol for the radio 8](#_Toc496995837)

[Overview 8](#_Toc496995838)

[Not connected 8](#_Toc496995839)

[ID conflict 8](#_Toc496995840)

[Measuring 8](#_Toc496995841)

[Standing still 9](#_Toc496995842)

[Wants to move 9](#_Toc496995843)

[Moving 9](#_Toc496995844)

[Beacon Mode 10](#_Toc496995845)

[Stopped moving but waiting 10](#_Toc496995846)

[Spreading out 10](#_Toc496995847)

# Block diagramm

# List of robots

## Data of the own robot

This list was too wide for this page so I split it up into two parts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID of robot** | **Brightness** | **Group size** | **I’m a beacon** | **Moved since measurement** |
| ID | Brightness | size | Yes/No | Yes/No |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **On IR visible robot with lowest ID** | | | **On IR visible robot with second lowest ID** | | | **On IR visible robot with third lowest ID** | | | **On IR visible robot with forth lowest ID** | | |
| ID | abs | ang | ID | abs | ang | ID | abs | Ang | ID | abs | Ang |

**ID of robot (1 Byte):**

When the robot starts up it sets its own ID to 1. Every ID should only occur one time and the Number of the ID should increase from 1 to the number equal to the size of the group. There should be no gabs between the ID numbers.

**Brightness (1 Byte):**

**Group size (1 Byte):**

The Group size is the number of robots which are connected via radio.

**I’m a beacon (1 bit):**

If another robot want to move towards this robot it need to light the IR LEDs up and isn’t allowed to move. This flag gets reset whenever the robot is a master and set by the other robots.

**On IR visible robot with (second/ third/ forth) lowest ID (3 Byte):**

Since we will only build 5 robots there will never be more than 4 robots visible at the same time. That way we can limit the amount of storage the list needs. If for example only one robot is visible on the IR the other spaces are filled with zeros.

## Data of the other Robots

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID of robot** | **Brightness** | **Moved since measure ment** | **On IR visible robot with lowest ID** | | | **On IR visible robot with second lowest ID** | | | **On IR visible robot with third lowest ID** | | | **On IR visible robot with forth lowest ID** | | |
| ID | Brightness | Yes/No | ID | abs | ang | ID | abs | ang | ID | abs | ang | ID | abs | ang |
| ID | Brightness | Yes/No | ID | abs | ang | ID | abs | ang | ID | abs | ang | ID | abs | ang |
| ID | Brightness | Yes/No | ID | abs | ang | ID | abs | ang | ID | abs | ang | ID | abs | ang |
| ID | Brightness | Yes/No | ID | abs | ang | ID | abs | ang | ID | abs | ang | ID | abs | ang |

# Pathfinding with A\*

When the robot wants to move to a robot at a brighter place it doesn’t necessarily have to see said robot via the IR. In the List are all the connections between the robots stored. The robot calculates the path with these connections and the pathfinding algorithm. If the goal isn’t reachable because there is no connection then it searches for the next best robot.

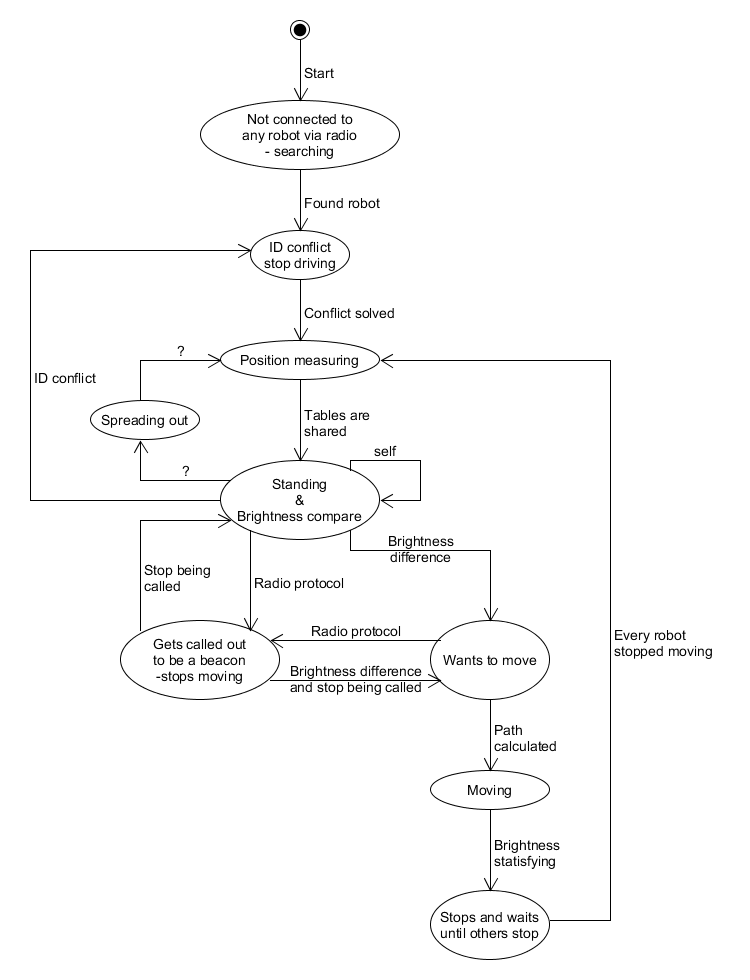
**I still have to figure out how to use A\*. Perhaps use depthfirst as an alternative.**

# Determining value of status LED

|  |  |
| --- | --- |
| Red | Digital out |
| Green | Digital out |
| Blue | Digital out |
| Delay | 1 Timer |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Blue**  Radio offline | **Green**  Idle state | **Red**  Too dark | **Yellow**  Driving to the light |
| If there is no robot in visible range (IR) it is always light up. |  | **X** | **X** |  |
| Blinks with 2.5Hz for the amount of robots in visible range (IR). It’s dark for a second afterwards. |  | **X** |  | **X** |
| While turning on spot it is 0.2 s on and 0.8s off. |  | **X** | **X** |  |
| Waiting for others doing something:  Diming on and of 0.5Hz |  | **X** | **X** |  |
| Measuring distance or other stuff |  | **X** | **X** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Flow cart



# Protocol for the radio

## Overview

The Starting info of every protocol are these three bytes.

|  |  |  |
| --- | --- | --- |
| IDown | Group size | Task |

IDown: Is explained in chapter [Data of the own robot](#_Data_of_the)

Group size: Is explained in chapter [Data of the own robot](#_Data_of_the)

Task: Refers to this [flow chart](#_Flow_cart). Will be mentioned in each of the protocols below.

Whenever the robot has sent all data which it wanted to send it ends the protocol with the byte

|  |
| --- |
| IDnext |

IDnext has normally the value IDown+1. If IDown has the same value as group size then IDnext is 1. Of course you can turn of a robot while the others are still running. If that is the case this robot won’t respond to the previous robot and the Robot sends an ID conflict.

## Not connected

Here the robot sends only the starting info. At the startup the ID is always 1 as well as the group size. The robot always switches between a longer period of random length of reading and sending a short message. As soon as it receives a message from another robot it goes into the ID conflict state.

|  |  |  |
| --- | --- | --- |
| IDown | Group size | Task |

Task: 0001’0000 (16)

## ID conflict

No clue jet how to solve that.

Task: 0010’0000 (32)

## Measuring

Here the goal is for every robot to know where which robot is. So every Robot lights up its IR LEDs one at a time. That way the surrounding robots know its position and the power consumption is low enough that the battery can support it.

When it is the robot turn, it first sends just the starting info.

|  |  |  |
| --- | --- | --- |
| IDown | Group size | Task |

Task: 0010’1000 (40)

After that it lights up all the LEDs one after the other. When it has finished with that it sends the Starting info again and its brightness and the ID of the next robot.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IDown | Group size | Task | Bright- ness | IDnext |

Task: 0010’1001 (41)

This step of the flow chart gets executed by every robot of the same group. When all the robots finished this task they make another round to share their measurement with the following protocol.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IDown | Group size | Task | Number of IDs | ID1 | abs1 | ang1 | ID2 | abs2 | ang2 | … | Zero | Bright- ness | IDnext |

Task: 0010’1010 (42)

IDn, absn and angn: transmit the data of the visible robots. The content of the data can be found in chapter [Data of the own robot](#_Data_of_the).

Zero: Zero is a byte filed with zeroes. It is there to show the receiver that the list of visible robots has ended.

## Standing still

In this step the robots basically only share their brightness and show that they are still active. That is done with the following protocol.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IDown | Group size | Task | Bright- ness | IDnext |

Task: 0011’0000 (48)

## Wants to move

As soon as there is another robot at a much brighter spot this robot wants to move to the brighter one. The A\* Algorithm calculates the shortest possible path to the other robot.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IDown | Group size | Task | Bright- ness | IDnext |

Task: 0100’0000 (64)

## Moving

Shortly after the path to a brighter spot is calculated the robot starts to move. It also tells the other robots where it wants to move to and which robot it has to follow to get to that destination. The robot always moves to the left most visible robot on the protocol. All the robots right to that on the protocol get removed from that list. All the robots in the List aren’t allowed to move because then the position after moving isn’t known anymore.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IDown | Group size | Task | Number of IDs | IDdestination | IDsecond last | IDthird last | … | Zero | Bright- ness | IDnext |

Task: 0100’0001 (65)

Zero: Zero is a byte filed with zeroes. It is there to show the receiver that the list of visible robots has ended.

## Beacon Mode

When in beacon mode the robot is not moving. This step has priority over the step [robot wants to move](#_Wants_to_move) and [standing still](#_Standing_still). As long as a robot that wants to move or is moving has this robot on its protocol, this step is active for this robot. The other robots must know where this one is and therefor this robot lights up its IR LEDs one at a time. As soon as it isn’t on the protocol anymore it stops the beacon mode.

Before the LEDs light up this protocol gets send via the radio.

|  |  |  |
| --- | --- | --- |
| IDown | Group size | Task |

Task: 0100’0010 (66)

After that this protocol gets send.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IDown | Group size | Task | Bright- ness | IDnext |

Task: 0100’0011 (67)

## Stopped moving but waiting

When a robot moves the position has to be measured again afterwards. When several robots are moving the position can only be measured again when all robots stopped moving again. The robots which have finished moving and are waiting for the others are in this step. When all robots have finished they change into the step [position measuring](#_Measuring).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IDown | Group size | Task | Bright- ness | IDnext |

Task: 0100’0100 (68)

## Spreading out

No clue jet how to solve that.

0101’0000 (80)