**AUTONOMOUS ROOM MAPPING ROBOT**

**Using image processing**

**Components:**

1. **Arduino Atmega328P (Arduino Uno)**
2. **Raspberry Pi**
3. **Camera**
4. **ZigBee**
5. **Servomotor**
6. **Distance sensors**
7. **Chassis**

**Software:**

1. **Arduino IDE**
2. **Python**
3. **OpenCV library**
4. **Matlab – maybe**

Arduino will be used for controlling the motors and reading the information from distance sensors. This information are sent from Arduino to Raspberry Pi.

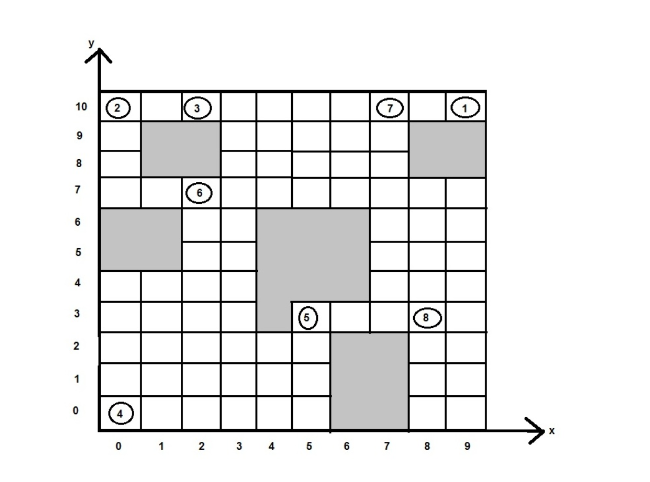
Raspberry Pi will read the information from camera and will perform the image processing algorithm, in order to detect the obstacles and walls.

The communication between Arduino and Raspberry Pi will be realized using Serial communication

(USB cable).

ZigBee module will be used for sending the information from Raspberry Pi to PC.

Matlab will be used for matrix generation and generating the map.



The project will be splitted in two parts:

1. Scanning and mapping
2. Navigation

For the scanning process, the robot will detect the walls and obstacles using camera and also, by making use of distance sensors will detect the distance to the obstacles. The room scanning process starts with 180˚ scanning: the robot will start from a corner of the map in which will take different samples at different angles. The samples will contain the distance to the obstacle and also the angle (the position of the servomotor). This samples / information will be sent to Matlab, where they will be converted from polar to cartesian coordinates. In this way, the data from the robot will be integrated into one single map and will be displayed on Matlab.

The second step of the scanning process is the matrix generation, where there will be three selected samples at 0, 90 and 180˚ as predefined values. The ranges determined for these angles will represents the blocks for the forward, left and right direction. The matrix will be generated by assigning ‘1’ to the free spaces (without obstacles) and ‘0’ to the obstacle’s blocks. The matrix would look something like this:

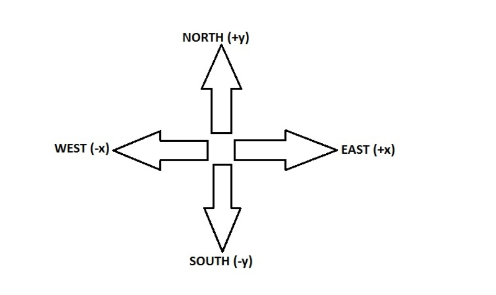
|  |  |  |  |
| --- | --- | --- | --- |
| Left | Center | Right | Block position |
| 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 2 |
| 0 | 1 | 0 | 3 |

Depending on the matrix, the robot will navigate the entire room by avoiding the obstacles and also map the room.

After returning to its original position, the complete matrix will be sent to Matlab.

For the navigation part a pointer will be defined, which will have the initial value of “north”. When the robot turns into a direction, the pointer will be incremented or decremented by one, depending on which direction is facing.

The (x,y) algorithm is used to calculate the current block position of the robot (x,y) as shown in the following picture:



Problems:

1. How do I know when the robot returns to its original position?

Based on the (x,y) coordinates or comparing the first set of values (from the image) with the current set of values (from the current image).

1. How the robot will know its position on the map?

The map will be splitted into multiple squares (or blocks) and these block will be mapped in the memory, based on the (x,y) position.

1. How will the robot map whole room?

The robot will be sent to read the information from all free blocks.