WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Ultrasonic Sensor Explorer

Prathamesh Suresh Malode*1,Prof.ChetanBundele*2

*1Student, Department of Mechanical Engineering, Prof Ram Meghe Institute of Technology &Research(PRMITR), Badnera, India.
*2Assistant Professor , Department of Mechanical Engineering, Prof Ram Meghe Institute ofTechnology&Research (PRMITR), Badnera, India.

ABSTRACT

Ultrasonic Sensor Explorer (USE) is a reasonably-priced cellular robotic for mapping. Instead of highly-priced laser sensor, it makes use of cheap ultrasonic sensor HC-SR04 with variety up to four meters. The chassis of the robotic is manufactured from ABS plastic fabric and it is 3D published from a version called Mini-Me on RepRap Prusa Iteration three three-D printer. The hardware is based on ROMEO BLE mainboard. The Software consists of components: firmware and software. The firmware element is an Arduino supply code venture, written in C++ programming language, which is deployed on the robot. The software element is Windows IOT source code, written in C# programming language, which may be run on PC, pill, telephone or ARM tool. The advised robothas been examined within the college lab surroundings. The provided idea proves it has a ability for SLAM.

INTRODUCTION

The mapping of the operating surroundings, addresses the hassle of acquisition of spatial models of the surroundings, necessary for making plans a protection direction from the starting position of the robotic to its goal vacation spot. The first rate diversity in the creation of maps of the operating scene [1] is determined with the aid of the one-of-a-kind techniques used for localization and movement planning of the robotic. Generally, the maps are divided into metric and topological. To decide the placement of the robots in metric maps, a hard and fast coordinate gadget is used. The topological map makes use of spatial relationships between items placed inside the working surroundings. The organization of metric maps consists of sensor maps, grids, geometrical and maps containing markers.

Maps, primarily based on sensor statistics [2], are characterised by the facts acquired from on-board sensors, which is stored without further processing. A regular example of a sensor map is a cloud of points, every of which is recorded via its coordinates in a global coordinate reference system. They are appropriate for applications related to archaeological studies or urban making plans, in addition to simulations and making plans the direction of the robot. For this reason, it is important, the 3-d cloud to triangulate to create a 3-dimensional picture [3], [4], [5]. The image may be similarly processed, if small occupied reminiscence is needed.

Grid maps, unlike other species, are one of the oldest and maximum popular maps in cell robotics [6], [7]. Through them, the environment is provided within the shape of a grid made from same cells. Each cellular is assigned with a vector, which describes certain bodily residences of the running area. If the cells incorporate records approximately the height of the terrain, theGeometric maps [15], [16], [2], [17], [6] used to symbolize the running environment, depict only critical quantities of the work scene, thereby their use overcomes the drawbacks, related to the inefficient use of computing resources at the robotic. Moreover, they do no longer discretize environments and consequently are greater correct in comparison with the grid maps. The modeling of environment turns into the idea of a set of geometric primitives, extracted within the shape of linear segments or polygons [6] in two-dimensional space and planes and in three-dimensional space. Geometric maps are appropriate for planning the motion of robots mainly in dependent and indoor environments. In comparison, outside environments are very complicated to be represented and modeled only with a few geometric primitives. The biggest gain of a geometrical map is its capacity to show the objects within the environment within the form of CAD fashions.

Maps, primarily based on markers [18], are a hard and fast of visible items with recognized positions. Unlike beacons, markers are not located into the environment for the motive of localization. The main cause of those maps is the localization of the robotic, but they'll also be beneficial for making plans the route of the robots. An crucial hassle of their use is the extraction of the most feature features of the float of facts with statistics from the sensors, which makes their use pricey in terms of the essential computational resource. During the development of the map, the recognized markers ought to be capable of speak with the tags, stored on the neighborhood map. Typically, these markers are recognized with the aid of snap shots received from the onboard digicam, but their extraction is viable from statistics, obtained from lasers, sonars, or radars. Typical examples of markers in a closed environment often are doorways, posters on and corners of the walls. In out of doors environment, the scenario will become even extra tough, due to the fact markers are greater hard to pick out and are a topic to modifications in the natural surroundings. Unlike the metric maps, topological maps display a graphical illustration of the operating environment [9], [1], [19], [20], [7]. They are used to depict geometric ratios of found gadgets in the environment. For their presentation are used graphs, wherein vertices are the gadgets and ribs define the connectivity among objects. The major blessings and disadvantages of the topological maps, consistent with [19], are that their use permits an green manner of planning the robotic's path due to their simplicity, with which the working environment is represented, even as unique dedication of the placement of the robot isn't required. They also are useful in the symbolic representation of the running scene. These maps are not appropriate for the mapping of huge areas because of their difficult creation and updating, as well as the inability to pick out the best path to reach the goal destination. Combining metric and topological maps gives the accuracy of the metric and the compactness of topological maps inside the exploration of massive environments.

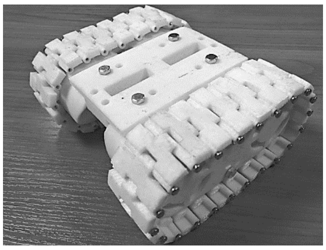
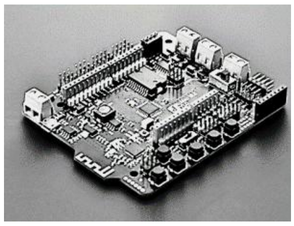
The intention of this paintings is to suggest a prototype of a cellular robotic for mapping. The robot can be used for obligations, such as testing and locating extraordinary simultaneous localization and mapping (SLAM) algorithms.

HARDWARE

Ultrasonic Sensor Explorer (USE)’s hardware part includes the following components:

- Ultrasonic sensor HC-SR04 with range as much as four meters, Shown on Fig. 1.A;
- Mainboard ROMER-BLE, Microcontroller: ATmega328P, Bootloader: Arduino UNO, On-board BLE chip: TI CC2540, Shown on Fig. 1.B;

Proto Tank Mini Me, made from ABS plastic on RepRap Prusa Iteration 3 3-D printer, Shown on Fig. 1.C

a/HC-SR04 b/ROMEOBLE c/ Mini-Me

ULTRASONIC SENSOR HC-SR04

The ultrasonic ranging module HC - SR04, as noted above, offers contactless dimension with 3mm ranging accuracy between 20 – 4000mm. The module includes ultrasonic transmitters, receiver and a control circuit. The fundamental principle of the work of this module is validated inside the next algorithm and timing diagrams proven in Fig. 2:

- Prepare HC-SR04 for measurements (ROMEO BLE mainboard trigger input pin for at the least 10µs excessive stage signal);
- ROMEO BLE mainboard watch for respond from module (the module automatically sends 8 pulses at 40kHz);

ROMEO BLE mainboard captures width of echo sign lower back from HC-SR04 (time of excessive output degree duration is the time from sending ultrasonic signal to it returning.

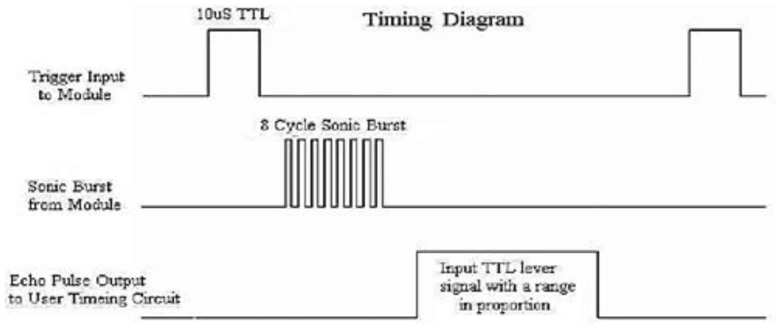


Fig. 2. Principal of work of HC-SR04 ultrasonic sensor

The assembled robot USE is shown in Fig. 3. The ultrasonic sensor is hooked up over the top of the chassis via micro servomotor.

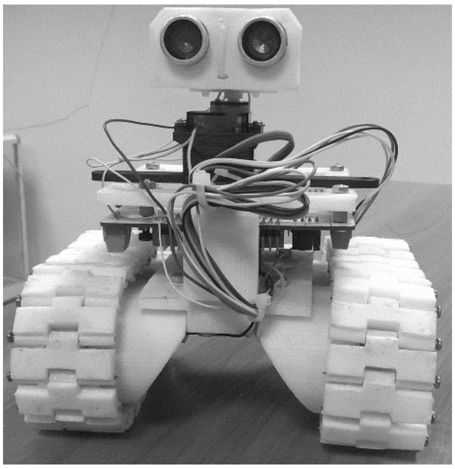
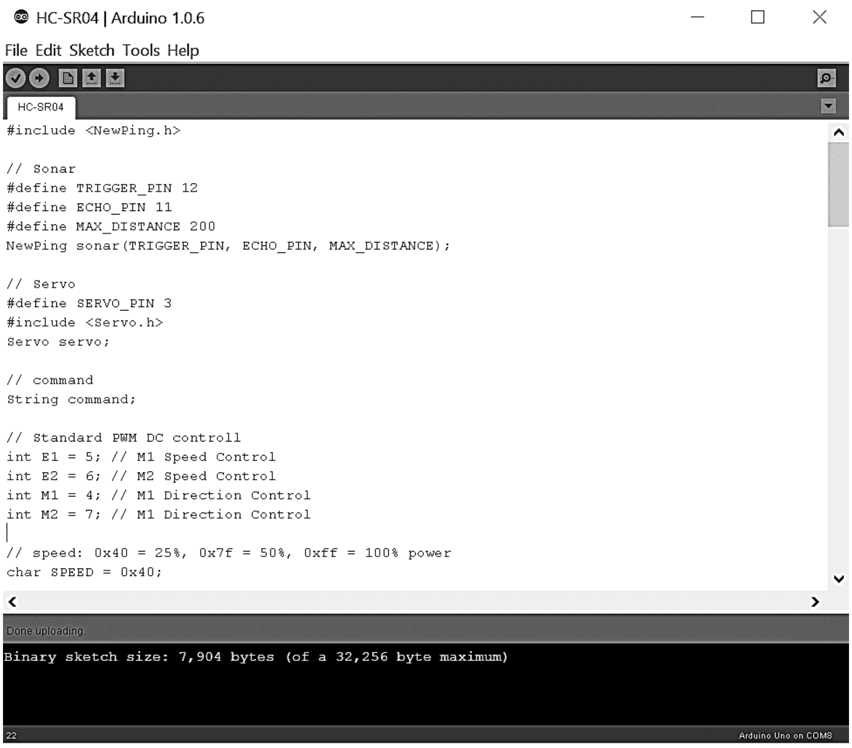


Fig. 3. Assembled robot USE

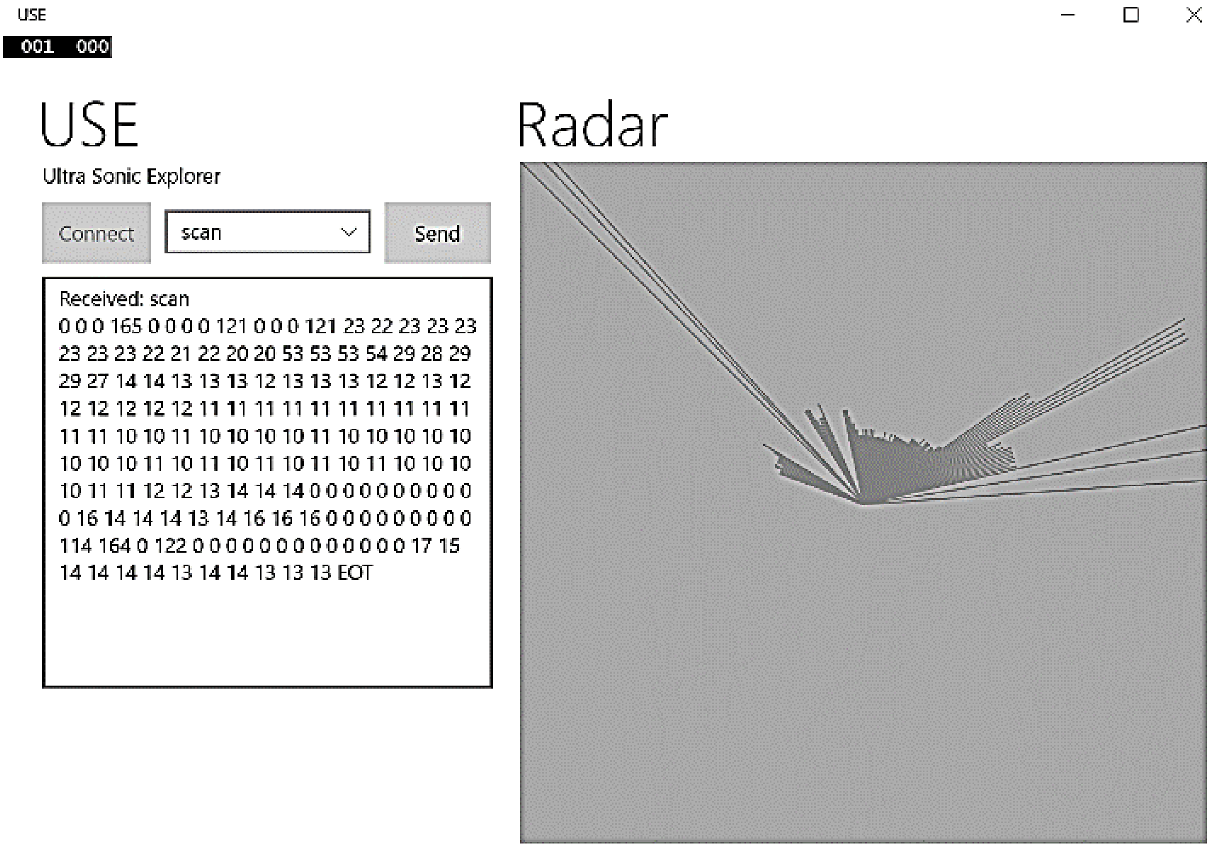
The cause of the micro servomotor is to rotate the sensor, at some point of scanning duration from the start line to the very last point and to return it to initial function for the next test.

SOFTWARE

Software consists of two separated parts: Firmware (Fig. Four.A) and Application (Fig. 4.B). The firmware component is Arduino source code mission, written in C++ programming language, that's deployed on the robotic. The application part is Windows IoT supply code, written in C# programming language, which can be run on PC, pill, telephone or ARM device



a/ Firmware



b/ Application

Fig. 4. Firmware and Application

The verbal exchange among these components of the Software: Firmware and Application, is offered on Fig. 5 The Application sends a SCAN command to the robot's firmware. When the robotic gets a SCAN command, the characteristic ProcessCommand() starts. This feature sets up the ultrasonic sensor HC-SR04 and micro servo motor to initial position. The startup procedure takes one second. In the following step, the firmware measures a hundred and sixty pattern probes, every one in all them takes 50 milliseconds to finish. During the technique of scanning, the robotic sends lower back every dimension to the Application. The Application is waiting and shops all the facts coming from the robot. At the end, the Firmware sends an End- Of-Transmission (EOT) command, which indicates that the approaches of taking samples is over. When the Application gets this command, it starts two features: ProccessDataBlock() and DrawControlPoints().

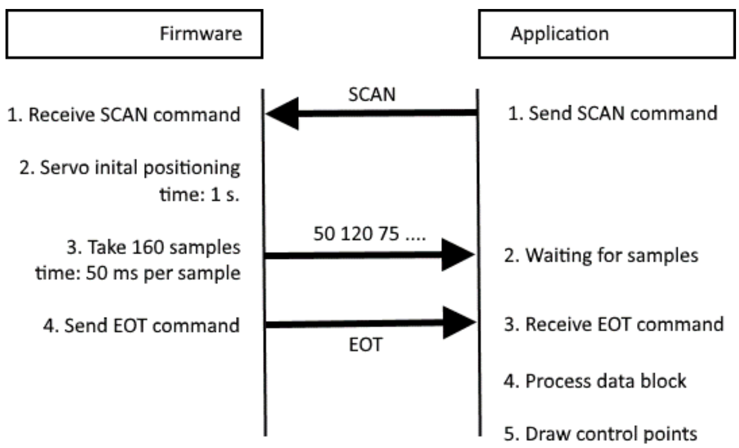
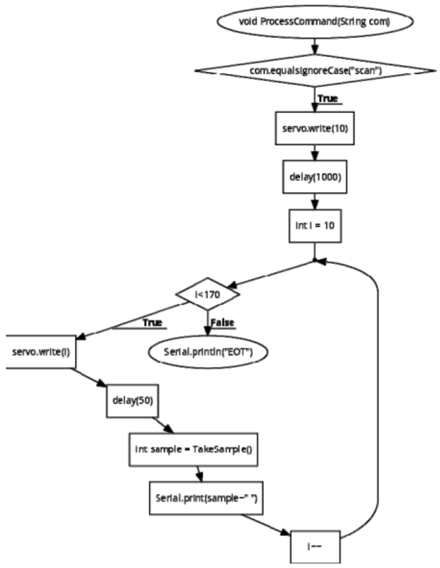


Fig. 5. Communication between software parts

The Firmware set of rules and supply code implementation for processing the acquired SCAN command is proven on the Fig. 6. The Application algorithms and supply code implementations for processing the obtained records block and visualizing calculated factors are proven on the Fig. 7 and Fig. Eight respectively.



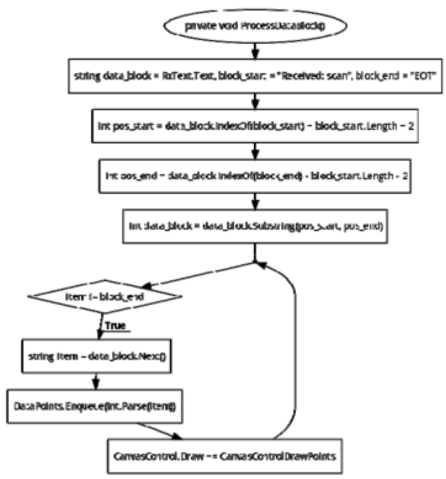
a/ Algorithm for processing the SCAN command

```
void ProcessCommand(String com)
{
   if (com.equalsIgnoreCase("scan"))
   {
      servo.write(10);
      delay(1000);
      int i = 10;
      while(i<170)
      {

      servo.write(i);
      delay(50);
      int sample = TakeSample();
      Serial.print(sample+" ");
      i++;
   }
   Serial.println("EOT");
  }
}
```

b/ Source code implementing the algorithm

Fig. 6. Firmware Partial Program Code



a/ Algorithm for processing the received data block

```
private void DrawControlPoints (CanvasControl
sender, CanvasDrawEventArgs args)
{
   Vector2 start = new Vector2(0, 0);
   Vector2 end = new Vector2(100, 100);
   Color color = Color.FromArgb(255, 0, 0, 255);
   double angle = Math.PI / 180.0;
   double step = Math.PI / 180.0;
   while (DataPoints.Count > 0)
   {
      int lenght = DataPoints.Dequeue();
      float endx = -lenght * (float)Math.Cos(angle);
      float endy = lenght * (float)Math.Sin(angle);
      angle += step;
      args.DrawingSession.DrawLine(new
Vector2(startx, starty), new Vector2(startx - endx,
starty - endy), color);
   }
}
```

b/ Source code implementing the algorithm

Fig. 8. Application algorithm and source code implementation for visualizing points calculated from the received samples

CONCLUSION AND FUTUREWORK

I. In this paper, a prototype of a mobile robot suitable for mapping is offered. It uses an ultrasonic sensor HC-SR04, which is a reasonably-priced opportunity for sensor mapping as opposed to an high priced laser sensor. The hardware is based on ROMEO BLE mainboard. The Software includes parts: firmware and application. The firmware element is an Arduino supply code undertaking, written in C++ programming language, which is deployed on the robotic. The utility component is Windows IoT source code, written in C# programming language, which may be run on PC, tablet, cellphone or ARM tool. The counseled robot has been examined inside the university lab surroundings.

II. Ultrasonic Sensor Explorer can be progressed by way of adding actual time records visualization. This can be done within the Application component, all through the reception of the pattern probes. Further in the software program can be carried out an set of rules for visualizing most effective the factors, whose values has been modified for the reason that remaining scan. This would cause minimizing the calculation and may produce a lighter prototype version of the robot for further exploration.

REFERENCES

[1] S. Thrun, „Robotic Mapping: A Survey,“ в Exploring artificial intelligence in the new millennium, G. Lakemeyer and B. Nebel, Ред., San Francisco, Morgan Kaufmann Publishers Inc, 2003, pp.1-35.

[2] T. Krajník, „LARGE-SCALE MOBILE ROBOT NAVIGATION AND MAP BUILDING,“ Prague,2011.

[3] D. Hahnel, M. Montemerlo, D. Ferguson, R. Triebel, W. Burgard and S. Thrun, „A system for volumetric robotic mapping of Underground Mines,“ in 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422),2003.

[4] J. Larson, B. Okorn, T. Pastore, D. Hooper and J. Edwards, „Counter tunnel exploration, mapping, and localization with an unmanned ground vehicle,“ in SPIE 9084 Unmanned Systems Technology XVI,2014.

[5] S.Thrun,S.Thayer,W.Whittaker,C.Baker,W.Burgard,D.Ferguson,

[6] D.Hahnel,M.Montemerlo,A.Morris,Z.Omohundro,C.Reverteand

[7] W. Whittaker, „Autonomous exploration and mapping of abandoned mines: Software architecture of an autonomous robotic system,“ IEEE Robotics and Automation Magazine, vol. 11, № 4, pp. 79-91, 2004.

[8] M. Veeck и W. Veeck, „Learning polyline maps from range scan data acquired with mobile robots,“ 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), vol 2,2004.

[9] С. А. Нойков, „Апаратни и програмнисредствазаобработканасензорнаинформацияприизследваненасредачрезкласмобилнироботи,“дисертация,2003.

A. Birk, K. Pathak, W. Chonnaparamutt, S. Schwertfeger, I. Delchev and S. Markov, „RoboCupRescue - Robot League Team IUB Rescue, Germany,“2006.

[10] F. Bonin-Font, A. Ortiz and G. Oliver, „Visual Navigation for Mobile Robots:ASurvey,“JournalofIntelligent&RoboticSystems,vol.53,

[11] № 3, pp. 263-296, 2008.

[12] J. Cox, „Blanche--An experiment in guidance and navigation of an autonomous robot vehicle," IEEE Transactions on Robotics and Automation, vol. 7, № 2, pp. 193-204,1991.

[13] G. N. DeSouza and A. C. Kak, „Vision for Mobile Robot Navigation: A Survey," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol 24, № 2, pp. 237-267,2002.

A. Elfes, „Using Occupancy Grids for Mobile Robot Perception and Navigation," Computer, pp. 46-57, June1989.

[14] Konolige, „Improved Occupancy Grids for Map Building" Autonomous Robots, vol. 4, pp. 351-367,1997.

A. M. Yamauchi, „PackBot: a versatile platform for military robotics," in Defense and Security,2004.

[15] O. Arras and R. Y. Siegwart, „Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building," Proceedings of SPIE, Mobile Robotics XII, vol, 3210, pp. 42-53,1997.

[16] H. Baltzakis and P. Trahanias, „An iterative approach for building feature maps in cyclic environments," IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1,2002.

[17] D. Sack and W. Burgard, „A Comparison of Methods for Line Extraction from Range Data," In Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV), № 1973,2004.

[18] G. Chronis and M. Skubic, „Robot navigation using qualitative landmark states from sketched route maps," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, vol. 2, № 1,2004.

[19] S. Thrun and A. Bucken, „Integrating Grid-Based and Topological Maps for Mobile Robot Navigation," in Thirteenth National Conference on Artificial Intelligence, Portland, Oregon,1996.

[20] S.Thrun,J.-S.Gutmann,D.Fox,W.BurgardandB.J.Kuipers,

[21] „Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach," In Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence, 1998.

[22] S. Thrun, „Learning metric-topological maps for indoor mobile robot navigation," Artificial Intelligence, vol. 99, pp. 21-71,1998.

[23] Ultrasonic Sensor HC-SR04, http://www.micropik.com/PDF/HCSR04.pdf

[24] Proto Tank Mini-Me,http://www.thingiverse.com/thing:1294208

[25] RepRap Prusa Iteration 3,http://www.reprap.org/wiki/Prusa_i3

[26] ROMEO BLE,

[27] http://www.dfrobot.com/index.php?route=product/product&product_id= 1176&search=romeo&description=true#.VrxgX49OKsc