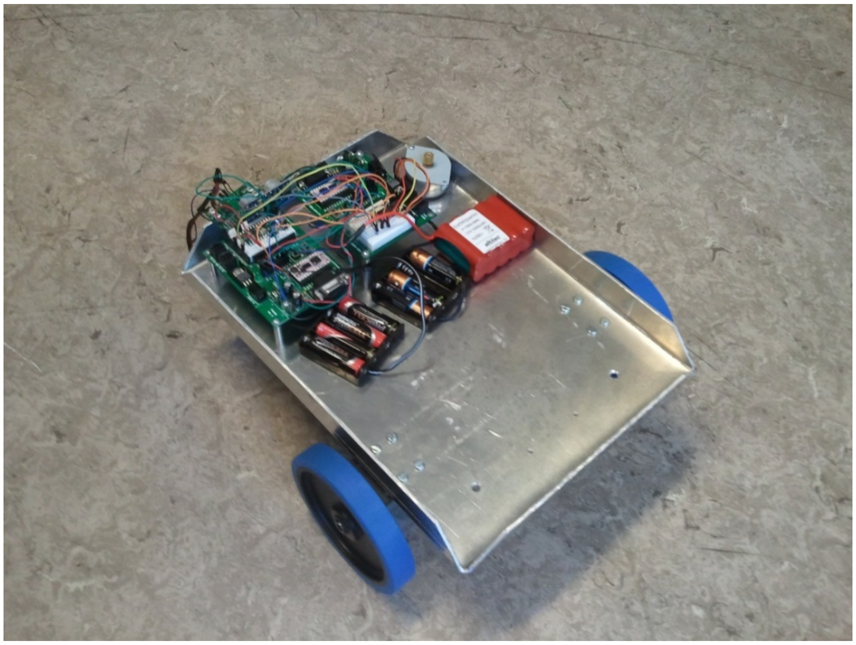
Prototype robot



Aso Abdulkarim Nizamaldeen

٣٠/١٠/٢٠١٣

Innhold

١. Summary

This Project is about designing a robot for different prototype purposes. The final goal is to use this robot as a prototype for a house-mapping robot.
The robot has different sensors and actuators with user friendly programming functions and is generally easy to program. The robot has two different chips that are connected in a "master slave" fashion.

Most of the early set goals of this project were achieved, but some work still remains doe to failing components and components refusing to work together.

The challenges consisted for the most part of compatibility problems and technical errors which couldn't be fixed. The compatibility problems we fixed by programing new sets of libraries, and the components with technical problems can be replaced.

٢. Introduction

٢.١ The goal of the thesis

It was difficult to find a definition of our project. There were many decisions that had to be made without enough data to support them. But what was clear was how the end product was going to be.

The end product was going to be a robot that could eventually map a whole room and then use this map to perform tasks on the room, i.e. vacuuming the room or sending the collected data to a computer for further analysis. This meant that the task was to find out what was needed to create such a robot and to detail what the difficulties are and find possible solutions.

The robot should also be easy to modify. This is done by creating a chassis with having room for future add-ons and modifications. Also the library is written in such a way that it is easy to reprogram and add to. All sensors and actuators are bought from parallax to ensure compatibility.

٣. Definition of the project

٣.١ Specifications

The robot is a prototype robot that generally can be used to collect data from an indoor room. The robot can then perform tasks based on those data collected. The tasks to be performed are programmed by any programmer with medium skills in object based programming.

The programmer of this robot is going to program the robot based on its microprocessors, Javelin Stamp from Parallax, with our libraries specially tailored for this robot.

The different sensors are:

- ٣ SONAR,
- Laser distance measurement,
- Gyroscope
- Position controller
- Compass

Of actuators are there:

- Step per motor
- DC-brush motor
- The possibility to insert servo motors

Another goal of the thesis is to create a prototype robot for laboratory use in school, where students can program and build upon.

٣.٢ Requirements

-The robot should be easy to program with Javelin Stamp IDE
-Library tailored for the specific sensors and actuators.
-Time used on project is limited to the spring semester of ٢٠١٢
-The budget is not to exceed ٦٠٠٠ kr.

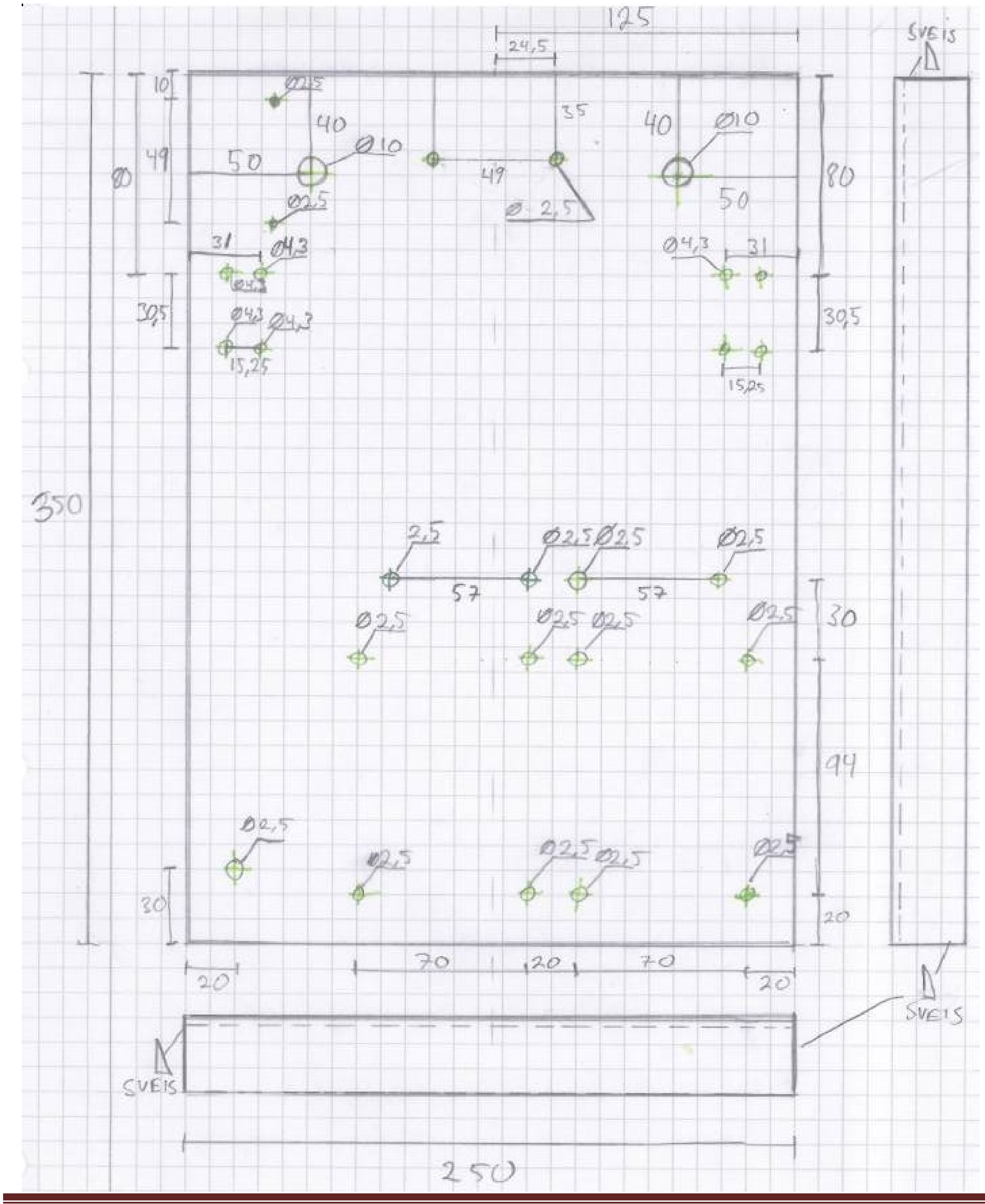
٤. Analysis

The project is divided into three parts; Design, programming and wiring. In the following chapter we will go through all three parts in succession.

٤.١ Design

٤.١.١ Chassis

Figur ٢, Chasis



٤.٢ Microcontrollers

There are two microcontrollers chosen for the robot; Javelin Stamp (Javelin) and Basic Stamp ٢ (BS٢). The reason for this is that the BS٢ is more compatible with the sensors and actuators, while the Javelin is easier to program libraries for. The Javelin is acting as the master and the BS٢ as the slave.

The master microchip is going to contain the main program that the user has access to program easily from Javelin Stamp IDE. The master microchip has access to all the specially tailored libraries written during the thesis. It will also have control over some actuators which we believe will require a short response. We chose Javelin as master because of its excellent properties such that it can be pulse width modification and stopwatches in other threads. This could, for example, drive motors in a certain time while the main program runs in a different thread. It is also well suited for communication due to its UART protocol with buffer and ease of use.

The slaves shall be of the BS٢ and Javelin. This is because most sensors are more compatible with the BS٢, but Javelin is generally more practical and performance-wise. The slaves will control all the sensors and have easy communication with the master via UART.

٤.٣ Sensors

For more detailed information than this we refer to the appendix regarding sensors.

Ultrasonic sensor PING from Parallax is used to measure distances. This has fast measurement intervals and fine accuracy. It works for distances from ٢ cm to ٣ meters, and is very straightforward to check for obstructions or check whether there is such a wall on your left. This sensor is compatible with both Javelin and BS٢.

Laser-distance meter Laser Range Finder (LRF) is a more accurate way to measure distances from ١٥ to ١٢٢ cm. It has a given average accuracy of ٣% with a maximum error value of ٥%. This is suitable pin for more precise measurements, such as measuring distances to find the shape of an object robot around.

Position gauge is attached to the robot's left wheel to determine how fast the wheel spins or how far the robot has traveled. On our robot's accuracy meter gauge with about .... and this is high enough accuracy for general prototype use.

The gyroscope measures the change in angle, or more accurately, the angular velocity. This measurement we use to prevent the robot pulls to one side. We also integrates angular velocity with trapezoidal rule to determine the angle of the robot has turned in functions that require this. The sensor requires a resolution of approximately one step per degree change.
We will use the compass to ensure that the angle we turned right. This sensor is quite disturbed in the indoor environment, but we think it works well enough for our use. It can also be used in many other ways for navigation.

٤.٤ Actuators

To drive the robot forward, we use two DC brush motors. The secant be from٦ to ١٢volts anddrawsabout٠.٢٥ampsat most. The wheels we got the engines slipping and will not cause the position and operation sensor works well. Top speed is about ٢m/s which is a decent rate for us.

We use a motor controller for controlling the DC brush motors. This can tolerate fairly high peak power, and we are well below the maximum current in normal operation. The controller can determine how fast and in what direction the wheels will go.

Currently it is not mounted some servo motors on the chassis, but we have nevertheless made library functions that make it easy to use servos.

Stepper motors, we have also created programmatically support. We use this to turn on sensors that need turning such as distance sensors. These have٧.٥ degree resolution is high enough for our use and easy to program.

٤.٥ Programming level

Here we divide the task into three parts: one master and two slaves. The master must include the main program that the user program using the Javelin Stamp IDE. This has access to all the libraries we've written. The master shall be a microcontroller of a Javlin Stamp. The master will also have control over some actuators which we believe will require short response time.

The slaves will be two different microcontrollers, a Javelin Stamp (Javelin) and a Basic Stamp ٢(BS٢). BS٢ will gain control of the sensors that we did not get to program, because of incompatibility, or did not fit on the Javelin. JavlinandBS٢ will communicate with the master through each UART. Javelin will have control over some actuators. For more detailed programming procedures, see attached file Robot.cbf.

٤.٦ Programming

Javelin is used both to one of the slaves and the master. One program this in Java using the Javelin Stamp IDE. Java is very similar to C#, so we did not think it was a difficult transition even if we could somehow java before.

The language on the BS٢ is called the Basic Stamp and this we could already because the previous lab assignments. This is a fairly easy language to learn because it is so easily constructed.

The communication should work properly, we had to create new function sin UART library. We made one to send١٦-bit inte organize and to receive it. This is a lot better than send in gone byte and then manipulating them to be further narrower. Although it was a little difficult to get UART protocol to work with a standard serial connectionbetweenJavelinandBS٢, so it went in the end. The slaves are programmed so that they would have the lowest response time. It turns out that this is between ٥٠ and١٠٠ milliseconds. They are programmed by the first initsierer all sensors and actuators that they have access to, and then they go into an infinite loop where the first reading input from the master and then use a switch to determine which metrics slave to do.

Here is the C-pseudo code for slaves: (TO JMJ: We know that it will be of code in the document itself, but is pseudo-code okay?)

```
Main {
int input;

        initiate();

        while(true){
                input = getInput();
                switch(input){
                        case ١:
                        sendOutput(measureOne());
                        break;
                        case ٢:
                        sendOutput(measureTwo());
                        break;
                        …
                        default:
                        pause(١٠);
                        break;
                }
        }
}
```

Here is the number cases and measurement function in the chosen based on which sensors are connected to the slave.

The master has access to all the library code we have written, and will use the slaves to perform measurements.

Test programs we have used come from parallax their web sites. Most of these are for BS٢, but not Javelin. Javelins test programs we wrote therefore, even where this was possible. The programs usually goes out on simple tests where they take a measurement and continuous writing this out to the Javelin Stamp IDE or Basic Stamp IDE. Test programs are attached.

In order to secure table ride we have implemented a P-controller to regulate both the speed and course.

٥. Cables & connections

Here is a description of how everything is connected together:

UART: The master is connected to the Javelin slave with four wires: two transmit/receive and two CTS/RTS. Transmit / receive is connected between pin ٣ on the master and pin ٠ on Javelin-slave and the master pin ٢ and pin١ on the Javelin slave. CTS/RTS is connected between pin١٠ on the master and pin١٤ on the Javelin slave and the master pin١٥ and pin١٥ on the Javelin slave.

The master is connected to BS٢slave with two links. Each of these has a resistance of at١٠kΩ.The links are from pin ٠ on the master to pin١ on the BS٢-slave and from pin١ on the master to pin٠ on the slave.

PING ١: is connected to the Javelin slave. Vdd to ٥V and Vss going to GND. Pin١١to the SIG.

PING ٢: is connected to the Javelin slave. Vdd to ٥Vand Vss going to GND. Pin١٢ to the SIG.

PING ٣: is connected to Javelin slave. Vdd to ٥V and Vss going to GND. Pin١٣ to the SIG.

Motor driver: is connected to the master. Pin٤ to"m١ in١", pin ٥ to"m١ in٢", pin ٦ to"m٢ in١", pin ٧ to"m٢ in٢", pin ٨ to"motor١ PWM" and pin٩ to"m٢ PWM". "M٢ non-PWM" and "m١ non-PWM" must be kept low with a link to Vss or GND. EN must be kept high with a link to Vdd.

M٢out١and out٢m٢ to be connected to one DC motor and m١out١ and m١out٢ be connected to the other.١٢V battery should be connected to GND and Vin on the same side of the chip as the other links listed in this section

Stepper: is connected to Javelin slave via a ULN٢٠٠٣ or similar IC. Phase one is connected to the١Cphase٢ to٢Cetc. Common is connected to the comULN٢٠٠٣. E is connected to ground. The colors of the phases can be found in the data sheet are attached. There are also illustrations to the layout.

The link between ULN٢٠٠٣ and Javelinslave is as follows: pin٢ to١B, pin ٣ to٢B, ٣Btopin ٤ and pin٥ to٤B.

The battery should be connected to the com/ common and earth.

The first stepper motor must be mounted with laser distance meter, but a second stepper motor can be connected in the same way. Then all Javelin pin be four times higher.

Position counter: Are connected toBS٢with pin١٣it is connected to DATA. VDC is connected to Vdd and GND to Vss.

Laser distance counter: Is connected to BS٢ with two cables. S out connected to pin٣ an pin٤toSin Vcc to the Vdd and GND to ground.

Gyroscope: Is connected to BS٢ with three cables. Dout (pin ٢ of the gyroscope) is connected to pin ٦ofBS٢, Sclk(pin ٤ ofthe gyroscope) is connected topin٥onBS٢andCsn(pin ٥onthe gyroscope) is connected topin٧ofBS٢. For detailed explanations and drawings, see data sheet in the appendix.Pin٣ofthe gyroscope is connected to ground and pin ٦is connected to Vdd.

For summary of connections, see the appendix" Summary of connections".

٦. Testes

For each sensor /actuator we have , we have created or downloaded a test program to see that each sensor /actuator mal function. In addition, we have carried out tests to make sure the master, slaves and sensors/actuators work together.

٦.١ Test programmer

All sensors and actuators have at some point worked with its test program outside the compass sensor.

Compass sensor had such program designed BS٢p that supportsI٢C, butnotBS٢. Since JavelinhasI٢C functional it we've tried to create a program that gets Javelin and compass to work together, but the compass only returns-١whichcan show that the compass has damage. This is more likely something to do with the code.

Position control, laser distance sensor and gyroscope worked with test their code, but doesn’t any more. This is very strange since we have found no fault in them, bread board we use (Board of Education) or link seven If we are connected to different terminal board, burning program on several new microcontrollers and tested lines of fracture. When we got in touch with Parallax was their theory that there was surge in one of the circuits that have destroyed the sensors and this we did not.

The sensors/actuators that still work are stepper motor, DC motors, all micro-controllers, servos and the PINGul transonic sensors. These have all worked from the start and still works.

٧. Accuracy

We find that the accuracy of the different sensors varies.

PING usually works very well within its measuring range and provides easy to read results back. We have three such sensors, and they give very similar accurate measurements, though they vary. Measurements do not work on surfaces such angle that no sound waves coming back.

Laser distance meter was very accurate in its measurement range, but it has a drawback: it measures a maximum of once per second. This causes them in program must wait for a response, and frequent measurements are not available.

Gyroscope integration works pretty badly. I do not think this is a good solution, and would trade it out in time. If the robot turns١٨٠ degrees clockwise, then it should return the angle is zero, but there turns plus minus ٢٠ degrees. This is not acceptable for the robot operation.

Position meter works very well, and register any movement we need the measures.

٨. Challenges

٨.١ Chassis

The plan was to create a chassis made of metal which we bentto. The challenge of the metal plate we had originally chosen was that it was too thin to be able to keep the structure stable. The metal plate we had chosen was ٠.٩ mm steel. It soon became clearly show that the weight of the sensors and the wheels were too much.

٨.٢ Battery

We had great difficulty battery. The only challenge was to find a battery that lastedlong enough and that could be easily recharged.

The battery that we ended up buying was a١٢V ١.٥ Ah NIMH. Maximum driving the robot is provided with this battery ٣ hours.

٨.٣ Faulty components

We had a big challenge with faulty Motor Driver component(MC٣٣٩٢٦). We spent several days trying to troubleshoot component thorough and detailed. Motor driver gets into including a PWM signal and then hold the desired speed of the wheels so that the master does not use more resources on it. It has two sets of inputs and outputs so that it can control two wheels simultaneously. The error that we tried to find made motor driver has not deliver a constant signal to one engine. So the speed of the wheel varied or stopped altogether. Eventually we found it most likely that it was error component. When we replaced component worked as it should.

٨.٤ Inaccuracy

It was considered by some inaccuracy in the equipment from the start because of inaccuracies in the given components. Laser distance sensor, for example, the maximum measurement distance of١.٢m and an accuracy of ٣%. Due to task our robot to do was not inaccuracy of a problem. The robot should not make any detailed work, but only Graven movements.

٨.٥ Incompatibility

As we worked with more and more sensors we came down a problem. Our initial thought was that to order parts from only one manufacturer would make our task easier, and that the parts would be compatible wave orm with each other. This turned out to be wrong. Parallax uses multiple suppliers who stock many different parts to them. Compatibility does not seem to be a primary focus of Parallax. It takes very long time to put into each individual data sheets and write code for each sensor. We also see that many components are not compatible wave form with each other, and therefore require proprietary driver features that are costly to make.

٨.٦ Javelins missing

There are some deficiencies in the Javelin. It cannot be determined with long or double variables and integrin is only ١٦ bits. These deficiencies did some library functions had to be done unintuitive, as it is ٣٦٠٠ degrees in a circle.

٨.٧ Basic Stamp ٢'s missing

After much testing and reading about the functionality of theBS٢, we see that this micro control designer has some deficiencies. It cannot communicate viaI٢Cand cannot communicate with a good number of sensors. We evaded these type sensors, but had to buy the compass sensor used as the BUS. BS٢ does not work either as an effective master because it cannot run anything in a background thread, and cannot, for example driving servos or DC motors without stopping all other stream. It cannot communicate during these operations, and this is essential for some types of prototypes.

٩. Conclusion

We have achieved some of the goals of the task, but not all:

-Easy reprogrammable with Javelin stamp IDE
Done.

Library-adapted sensors and actuators:
We have only partially tested the library because the sensors have been in order. If we look at what we got before they were out of order, we are quite confident that the program code is correct, and the program will work fine for a complete sensor is connected.

-Sensors: ultra sonic sensor, laser distance sensor, gyroscope, position controls and compass
All sensors except the compass is in place. This goal completed is also limited by the fact that some sensors currently not working.

-Actuators, stepper motors, DC brush motors and the ability to connect to the servos.
Done.

-Limited time for spring semester٢٠١٢.
Done.

-The budget must be kept with in certain limits.
Completed, but comes at the budget because of parts that needed to be replaced and extensions of sensor range.

٩.٢Recommendationsfor further work
Based on our results, we see that there are good opportunities for further work. We therefore setup certain work packages, each of which is a good upgrade of the robot.

١ **SensorRepair:** Sensors should be replaced. These need to troubleshoot the installation of bike. Then they must unite with already prepared features so that they achieve the desired outcome.

٢ **Programming a robot to serve as indoor mapping:** We have every thing prepared most of the functions needed, so when work package١ is finished, it is relatively easy to perform work package No. ٢

٣ **Otherprototype:** Robot intended use is as proto types. We see that it is difficult with the number of sensors that are able today after work package١ is completed it will be possible to use the robot proto typing for such garbage picking, cost robot ,rover, mazesolvesmm

٤Physicalextensionsof the robot:

We have also calculated the ample space of the robot, so if it is desired, attach as robot arms or more sensors on this. This can be done without completing the previous work packages.

١٠. Literature list

Java for nybegynnere:
http://thenewboston.org/list.php?cat=٣١
Date: ١٨.٤.١٢

Java for viderekomne:
http://thenewboston.org/list.php?cat=٢٥
date: ١٨.٤.١٢

Javelin Stamp manual:
http://www.parallax.com/Portals/٠/Downloads/docs/prod/javelin/JavelinManv١-٠.pdf
data: ١٨.٤.١٢

Basic Stamp manual:
http://www.parallax.com/Portals/٠/Downloads/docs/prod/stamps/BS١nBS٢ConvTips.pdf
date: ١٨.٤.١٢

Posisjonskontroller datablad:
http://www.parallax.com/Portals/٠/Downloads/docs/prod/motors/٢٧٩٠٦-PositionCtrlrKit-v١.١.pdf
date: ١٨.٤.١٢

Laser-distansemåler datablad:
http://www.parallax.com/portals/٠/downloads/docs/prod/sens/٢٨٠٤٤-LaserRangeFinder-v١.٠.pdf
date: ١٨.٤.١٢

Ping ultralydsensor datablad:
http://www.parallax.com/Portals/٠/Downloads/docs/prod/acc/٢٨٠١٥-PING-v١.٦.pdf
date: ١٨.٤.١٢

Steppermotor datablad:
http://www.parallax.com/Portals/٠/Downloads/docs/prod/motors/Stepper_Motor_٢٧٩٦٤.pdf
date: ١٨.٤.١٢

Gyroskopdatablad
http://www.parallax.com/Portals/٠/Downloads/docs/prod/sens/٢٧٩٢٢-LISY٣٠٠GyroscopeModuleV١.٠.pdf
date: ١٨.٤.١٢

DC-motor datablad:
http://www.parallax.com/Portals/٠/Downloads/docs/prod/robo/٥٧٠-٠٠٠٧٠-
٧.٢vMotorBracketWheelKit-v١.١.pdf
date: ١٨.٤.١٢

Kompass datablad:
http://www.parallax.com/Portals/٠/Downloads/docs/prod/sens/٢٩١٣٣-
CompassModuleHMC٥٨٨٣L-v١.٠.pdf
date: ١٨.٤.١٢

Motordriverdatablad
http://www.parallax.com/Portals/٠/Downloads/docs/prod/motors/٢٨٨٢٠_Pololu%٢٠Dual%٢٠
MC٣٣٩٢٦%٢٠Motor%٢٠Driver%٢٠Carrie١.pdf
date: ١٨.٤.١٢