Design and Development of an Automated Guided Vehicle for Educational Purposes

Khosro Bijanrostami

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Mechanical Engineering

Eastern Mediterranean University
September 2011
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Mechanical Engineering.

Assoc. Prof. Dr. Ugur Atikol
Chair, Department of Mechanical Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Mechanical Engineering.

Prof. Dr. Majid Hashemipour
Supervisor

Examining Committee

1. Prof. Dr. Majid Hashemipour
2. Assoc. Prof. Dr. Fuat Egelioğlu
3. Asst. Prof. Dr. Hasan Hacışevki

ABSTRACT

An Automated Guided Vehicle (AGV) is a set of cooperative driverless vehicle, used on manufacturing floor and coordinated by a centralized or distributed computer-based control system. AGVs-based Material Handling Systems (MHSs) are widely used in several Flexible Manufacturing Systems (FMS) installations. One of the challenge in MHSs is how flexible and adequate is the utilization. The key issue of the flexibility of MHSs is the routing system. It should be designed in a way that can be easily modified to become adaptable to new or replaced machines.

The main focus of this study is to make an AGV with the convenient materials, simple and applicable routing system and more importantly reducing the cost and increasing the flexibility. For this propose an AGV is basically modeled and designed with CATIA software and developed with special specifications such as producing some parts by milling CNC when high accuracy was necessary. Moreover the flexibility of the system is improved employing three more sensors which make the plan more intelligent dealing with multi directional guiding paths. Also benefiting from the colorful paths the flexible is enormously increased due to simplicity of the nature the paint to be plant or removed. Finally the users are able to extend components, add new machines, define them and specify routs for new settings without disturbing the operations in process. This thesis also addresses key issues involved in the design and operation of AGV-based MHSs for the FMS section of CAD-CAM laboratory of Mechanical Engineering Department of Eastern Mediterranean University.

Keywords: Fuzzy logic, steering system, programmable logic control (PLC), flexible manufacturing system (FMS), material handling system (MHS)

ÖZ

İmalat alanında kullanılıp, merkezi veya dağıtılmış bilgisayar tabanlı bir kontrol sistemi tarafından yönlendirilen, sürücüsüz yardımcı araçlara; Otomatik Yönlendirmeli Araç (OYA) denir. OYA tabanlı Materyal Taşıma Sistemleri (MTS), Esnek Üretim Sistemleri (EÜS) dahilinde sık sık kullanılmaktadır. MTSnde en önemli etkenlerden biri kullanımın ne kadar esnek ve yeterli olacağıdır. Esnekliğin anahtar noktası yönlendirme sistemidir. Yönlendirme sistemi, yeni veya yeri değiştirilmiş makinalara göre kolayca adapte edilebilecek şekilde dizayn edilmelidir.

Bu çalışmanın odak noktası, elverişli malzemelerle bir OYA yapmak, basit ama kullanışlı bir yönlendirme sistemi üretmek ve en önemlisi maliyeti düşük tutup sistem esnekliğini olabildiğince yükseltmektir. Bu sebepten bir OYA CATIA programında basit şekilde modellenmiş, ve yüksek standartlarda hazırlanmıştır. Büyük hassasiyet gereken yerlerde, CNC dik işlem cihazına baş vurulmuştur. Esnekliği artırma amaçlı olarak üç ek sensör eklenmiş, böylece sistemin çok yönlü kılavuz yollarda daha başarılı olması sağlanmıştır. Kılavuz yolları renginden tanıma özelliği ile, yolun kolayca boyanıp tekrar silinebileceği akılda tutularak, sistem esnekliğine büyük bir katkı daha yapılmıştır. Son olarak, kullanıcılara bileşenleri genişletme, yeni makineler ekleme, bu makineleri tanımlama ve bu eklemelere göre işlemleri rahatsız etmeksizin yeni güzergah çizme kolaylığı sağlanmıştır. Bu tez içerisinde OYA tabanlı bir MTSnin, Doğu Akdeniz Üniversitesi Makine Mühendisliği Bölümü CAD-CAM laboratuvarının Esnek Üretim Sistemi kısmı için kurulumunda yapılması gerekenler de anlatılmaktadır.

Anahtar Kelimeler: Bulanık mantık, direksiyon sistemi, programlanabilir mantık kontrol (PLC), esnek üretim sistemleri (FMS), malzeme taşıma sistemi (MHS)

ACKNOWLEDGMENT

It would have been impossible for me to complete this work without the help of the people who have supported me financially, emotionally and morally. First, I would want to acknowledge my supervisor, Prof. Dr. Majid Hashemipour for his relentless efforts and supports in making this work a reality. The Department Chair Assoc. Prof. Dr.Ugur Atikol and all the lecturers and technicians of the department, but not limiting them only, have also in one way or the other given me selfless support .To them, I will always remain thankful. To crown it all, I would want to show my gratitude to my primary mentors who are my parents, sister and brother in low for their endless support in everything I do in life. I will remain always indebted to you for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURS

Chapter 1

INTRODUCTION

Automation has become the core of modern manufacturing so much so that, no company is able to survive in a competitive market without automating its operations. In fact the term automation basically refers to the use of computer and other automated machinery for the execution of tasks that human labor would otherwise perform. Automation is used to manage systems and to control processes, thus leading to reduce the necessity of human intervention.

Nowadays, manufacturers seek to implement methods of automation appropriate to their needs and purposes. Companies automate their activities for a variety of reasons. Increasing productivity is normally the main aim for companies desiring competitive advantages. Automation reduces human errors and improves the quality of output. Other reasons of automation include the presence of hazardous working environments and the high cost of human labor in such areas. The decision regarding automation is often associated with some economic and social considerations Ravazzi and Villa [1], Chadwick and Jones [2].

In order to have adequate automation, there are number of issues to be taken into the account. Depending on the product and area, the components of automation might be different, but there are some elements that must always be considered such as the field of automation, scale or size of the place that is going to be automated, and what level of flexibility is required.

One of the key components of automation in a manufacturing process is the MHS. This system is responsible for loading, unloading, moving or generally transporting any type of materials (raw material, work in process, and finished good) within and out of manufacturing cells such as warehouses, machines and assembly lines. MHS consisted of different components (i.e. Conveyors, AGVs, Robots, Automated Storage and Retrieval system (AS/RS)). Utilization of components, either individually or from combination point of view, is determined by its application or pre-assigned flexibility.

In this study, AGV is considered as the most flexible equipment of MHS. An AGV is a driverless transportation system used for horizontal movement of materials. On the other hand it is an unmanned vehicle, controlled and driven by a host computer, to carry out the required material movement in a manufacturing floor. AGVs can be used in both interior and exterior environments such as manufacturing, distribution, transshipment and (external) transportation areas. In manufacturing areas, AGVs are used to transport all types of materials related to the manufacturing process.

Since the introduction of AGVs, there have been two methods of steering namely; *close path* and *open path,* which are employed based on the application, area size, cost and etc.

In the close path steering system, a line embedded or buried on the ground and a sensor set at the bottom of the AGV detects the line and guides the AGV to follow the line. There are two types of line in such systems, namely colorful line and magnetic line. The latter is embedded and former is buried in the ground and Color magnetic detective sensors are the two employed types of sensors. In such steering system the paths are fixed therefore frequent changes of design are not adopted in to nature of such systems. Consequently flexibility and changeability of design is less likely.

Therefore open path steering system came through to cover the problems inherent in the former system in which there were no physical paths for AGV. However, there were some virtual pre-define ones on the controlling unit. Thus the paths could be changed without physically changing the system. In such systems there are two orthogonal lasers rows. Crossing these orthogonal laser lines, the workshop becomes like a grid area. Using grids, paths are assigned in the supervisory control. Therefore the location of the AGV and also the probable deviation is controlled and supervised by the supervisory unit. In this steering system, although the accuracy and flexibility are high, it is much more costly compared to the close path steering method.

The proposed model is based on the close path system and introduces a unique movement for AGVs and steering model for the system. This has the main advantage of conventional close path method with fewer disadvantages in employing technical decision, making processes and heuristic algorithm for the steering system with specific hardware and software design.

In this method there is a colorful path from the initial position (home) to all of the stations. This path consists of a main path and three side paths, originating from the main path. Each side path ends to a station. And a perfect supervisory system is employed to control all the movements.

The examining area of the proposed AGV is the CAD/CAM laboratory of Mechanical Engineering Department of Eastern Mediterranean University. It is an experimental/educational shop floor in which there exist an oval flexible conveyor, integrated with another belt conveyor and three robots of which one of them is considered as conveyor station. These equipments are connected to a PLC and a computer called host computer. There exist two milling and turning CNC machine which are not connected to the system and consequently are not integrated.

The first chapter introduces the necessity of employing a FMS and advantages of using AGVs in those systems. It also introduces the subjective environment of the proposed AGV with a brief explanation of steering model.

The second chapter will review the history and back ground and the latest improvement of the AGVs in some aspects.

Chapter three will show the hardware design and all the component of the AGV. It will analyze of all the possible movements and also introduces some part of the software design dealing with hardware components.

Chapter four will illustrate the steering method including algorithm, dispatching rolls, deviation classification, deviation control, station recognition, idling and interface controllers.

The fifth chapter will discuss the results and reliability of the plan graphically and mathematically and also provide the best adjustments and specifications for the AGV to reach the goals under its specific conditions.

In the end, it is expected that all equipment should be integrated with the aid of the host computer of the system and the proposed AGV.

Chapter 2

LITERATURE REVIEW

Automatic Guided Vehicles (AGVs) have played a vital role in moving material and product for more than 50 years. The first AGV system was built and introduced in 1953. It was a modified towing tractor that was used to pull a trailer and follow an overhead wire in a grocery warehouse. By the late 1950's and early 1960's, towing AGVs were in operation in many types of factories and warehouses.

The first big development for the AGV industry was the introduction of a unit load vehicle in the mid 1970s. This unit load AGVs gained widespread acceptance in the material handling marketplace because of their ability to serve several functions; a work platform, a transportation device and a link in the control and information system for the factory.

Since then, AGVs have evolved into complex material handling transport vehicles ranging from mail handling AGVs to highly automated automatic trailer loading AGVs using laser and natural target navigation technologies.

In fact the improvement of AGVs over the last decade is deeply indebted to development of Scheduling, Algorithm and Steering methods. The problem of scheduling of AGVs and the other supporting equipments has been extensively studied by Basnet and Mize [3] and Rachamadugu and Stecke [4] currently providing the most up-to-date and comprehensive reviews in this area.

Han and McGinnis [5] have developed a real time algorithm in which material handling transporters are considered. Schriber and Stecke [6] have shown how the additional consideration of the material handling system and limited buffers degrades the system performance. Sabuncuoglu and Hommertzheim [7, 8] have highlighted the importance of material handling and they compared several AGV dispatching rules. They have also shown how the buffer capacity can affect the performance of the system. Flexibility, which is a distinguishing feature of FMSs, has received an extensive amount of attention. Routing flexibility (i.e., alternative machines and processing routes) has been considered by Wilhelm and Shin [9], Chen and Chung [10], and Khoshnevis and Chen [11]. These studies have indicated that dynamic routing (i.e., a path determined dynamically during schedule generation) performs better than a preplanned routing. Rachamadugu et al. [12] have proposed a quantitative measure of sequence flexibility and have shown that perfect sequence Flexibility improves system performance. Similar observations have been made by Lin and Solberg [13]. In most work to date, tools, pallets/fixtures and their availability are not modeled adequately. A static allocation of tools is usually assumed in these studies.

However, some researchers have considered a limited tool magazine capacity and the changing of tools from central tool storage [14]. One purpose of this thesis is to develop an algorithm that can be used to investigate the research issues discussed above. This algorithm should not only consider the major elements of FMSs but also generate high quality schedules in a reasonable amount of time. In this thesis, the basic structure and characteristics of such an algorithm is described.

Kim et al. [15] proposed a deadlock detection and prevention algorithms for AGVs. It was assumed that vehicles reserve grid blocks in advance to prevent collisions and deadlocks among AGVs. A graphic representation method, called the "reservation graph," was proposed to express a reservation schedule in such a form that the possibility of a deadlock can be easily detected. A method to detect possible deadlocks by using the reservation graph was suggested.

Maxwell and Muckstadt [16] first introduced the problem of AGV flow system design. While their main concern is vehicle routing, they also address material flow path and station location design issues. The flow network they used, known as *conventional configuration,* is composed of unidirectional arcs. Gaskin and Tanchoco [17] developed the first integer programming model for material flow path design. Given a fixed network of aisles and fixed pickup and delivery stations, the model assigns direction to arcs to minimize the total trip distances of loaded vehicles. Goetz and Egbelu [18] developed an alternative model, where the station locations no longer are fixed but restricted to the nodes on the boundary of the cells. Sun and Tchernev [19] provide a comprehensive review on the models developed for conventional configuration. Afentakis [20] states the advantages of the loop layout as simplicity and efficiency, low initial and expansion costs, and product and processing flexibility. Loop layout has been studied by many researchers including Bartholdi and Platzman [21], Sharp and Liu [22], Kouvelis and Kim [23], Egbelu [24], Banerjee and Zhou [25], and Chang and Egbelu

[26]. Bozer and Srinivasan [27] initiate the concept of tandem configuration as a set of no overlapping, bidirectional loops, each with a single vehicle.

Another problem in steering issues is to schedule several AGVs in a non-conflicting manner which is a complicated real-time problem, especially when the AGV system is bi-directional. In fact, many conflicting situations may arise such as head-on and catching-up conflicts when the AGVs or the guide-paths are bidirectional and if no efficient control policy is used to prevent them. Several conflict-free routing strategies have been proposed and can be classified into two categories:

- Predictive methods: Aim to find an optimal path for AGVs. The conflicts are predicted off-line, and an AGV's route is planned to avoid collisions and deadlocks [28-30].
- Reactive methods: the AGVs are not planned and the decisions are taken in a real-time manner according to the system state.

These methods are based on a zone division of the guide-path and consider them as non-sharable resources [31-33]. Predictive methods give good performance, but are not very robust since they do not take into account real time problems. However, reactive methods are very robust but the resulting performances can be poor because the decisions are taken by considering a very short-term time horizon [34, 35]. In this thesis due to specification of the whole plan (presence of only one AGV) a kind of predictive method is proposed.

In early 1990s Fuzzy logic came through to control and manipulate whole of the material flow in manufacturing floors. The main indication of employing this system on AGVs was the ability of controlling multiple AGV in a same time without collision.

Fuzzy logic is a very useful nonlinear control method that can be used to control very complex models or plants, which are formidable to model. Fuzzy linguistic terms allow the user to incorporate the heuristic knowledge of the plant in control law synthesis. Lakehal et al. [36] proposes fuzzy logic control for path tracking. It is based on position and orientation errors with respect to the reference path. Controlling two independently driven wheels achieves both the Longitudinal and Lateral control of the vehicle. However, only simulation results are presented. Senoo et al. [37] used experimental results of a three wheeled mobile robot to discuss the stability of a fuzzy controller. It is also stated that fuzzy control was implemented in order to achieve reduction of steer energy, while maintaining better steer angle when compared with PI control.

Fuzzy logic has found useful applications in control among other areas. One useful characteristic of a fuzzy controller is its applicability to systems with model uncertainty and/or unknown models. Another useful characteristic of a fuzzy logic controller is that it provides a framework for the incorporation of domain knowledge in terms of heuristic rules.

Wuwei et al [38]. They presented the new navigation method for AGV with fuzzy neural network controller when in the presence of obstacles. Their AGV can avoid the dynamic and static obstacle and reach the target safely and reliably.

Wu et al. [39] used fuzzy logic control and artificial potential field (APF) for AGV navigation. The APF method is used to calculate the repulsive force between the vehicle and the closest obstacle and the attractive force generated by the goal. A fuzzy logic controller is used to modify the direction of the AGV in a way to avoid the obstacle.

Lin and Wang [40] proposed a fuzzy logic controller for collision avoidance for AGV. They combined fuzzy logic with crisp reasoning to guide an AGV to get out of trap since memories of path and crisp sequence flows are handled by non-fuzzy processing. Their designed AGV was able to avoid collision with unknown obstacle.

Alves and Junior [41] used a step motor to turn the direction of the ultra-sonic sensors, so that each sensor can substitute two or more sensors in mobile robot navigation.

Perhaps Sugeno [42, 43] has done one of the pioneering researches in mobile robot navigation using fuzzy logic control. The fuzzy control rules, which he defines for the controller, were derived by modeling an expert driving action. He made a computer model of a car in microcomputer to find fuzzy rules. The speed of the designed car was constant; then, the control input to the car is only the angle of the steering angle

Mehdi Yahyaei [44] has design a AGV using fuzzy logic system and a rotational ultra sonic sensor to steer the AGV to avoid collisions and obstacles. He also employed a programmable logic control (PLC) as the processor which makes the AGV to be ultimately fit to the industrial environments.

Chapter 3

HARDWARE AND SOFTWARE

The procedure of designing an AGV is a complicated process. Some issues which directly impact the design of the proposed AGV are listed and widely explained. These issues are not only hardware but also software issues. Software is not just constants in inputs but it is variable and outputs must be chosen to specify the design. Furthermore, these issues interact with each other so that each cannot be considered separately but all must be considered simultaneously.

- Vehicle Hardware Design
  - Movement modeling
  - System Configuration
  - Kinematic Computation
  - Components of AGV
- Path and Guide-path Design
  - Interaction of path and sensors
  - Path's specifications
- Workstation Information
- AGV Scheduling
  - Idle-AGV positioning
  - AGV moving

  - Stopping (deadlock resolution)
- Software design
  - Algorithm
  - Programming
  - Interface commander
- Battery management

3.1 Vehicle Hardware Design

3.1.1 Movement Modeling

One of the most challenging parts of designing the AGV is the movement modeling. Movement modeling highly depends on the size of the area, expected maneuvering ability, position of stations and allocated path between them. Furthermore, it becomes much more vital if the area is small with restricted moving space so that the vehicle should be designed to move and make U-turns, sharp turns, curve turns and of course handling deviations.

Typically, standard AGV's are distinguished by having an axis in one rotational degree of freedom at the back and the front axis in two rotational degrees of freedom is to guide the vehicle on the arbitrary path. The most important advantages of such model are the accuracy and simplicity which make the plan highly applicable. However, there is a big disadvantage which is the weakness in maneuvering on the sharp turns. In fact in such models the minimum radios of the turn carve is restricted.

Despite of advantage of implementing standard models, another model has to be used. It is because of the specifications of the place for which the AGV is proposed. Due to the need of an AGV with the ability of any type of movement and turns, the military movement model is chosen. In this model there are 2 motors installed on each sides, connected to main controller kit. This kit receives the commands and respectively controls and drives each motor independently. It means that the vehicle is able to perform U-turn even on a point, which is the hardest and sharpest movement. Benefiting from this model, the vehicle is capable to perform all types of movements and turns. The following part will illustrate it with figures of all different capabilities of the vehicle.

3.1.2 System Configuration

The configuration of the vehicle forces is shown in figure 1. It is seen that the two sprockets in the middle of the AGV are driving wheels, which are actuated separately by two DC motors. So there are two trajectories of line and arc for this kind of AGV. The center of linking line between two driving wheels is the kinematics origin of AGV.

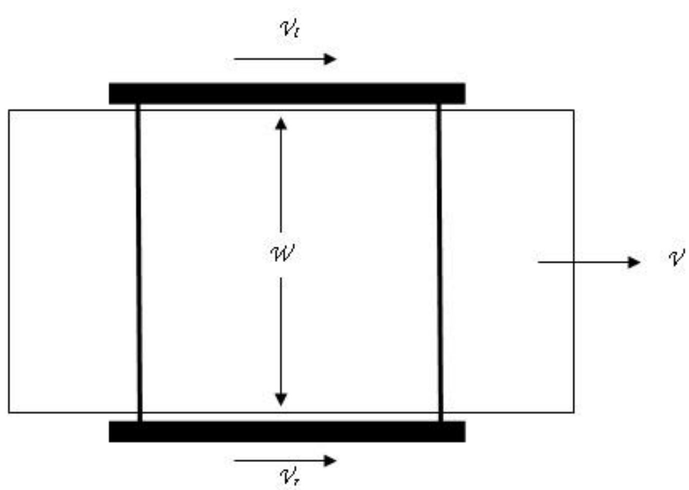


Figure 1: Vehicle Forces

V= linear AGV velocity (mm/s)

ω = angular AGV velocity (rad/s)

W = distance between two sprockets = 400mm

V_r = velocity of right wheel (mm/s)

V_l = velocity of left wheel (mm/s)

3.1.3 Kinematic Computation

In order to be able to configure the kinematic computation of the AGV, it is assumed that all of the belt's wheel force is applied to a point which is at the center of each wheel in each side.

Therefore for the linear movement, velocity is calculated as follows:

$$V = \frac{V_r + V_l}{2}\ (mm/s) \qquad (2.1)$$

And for the angular movement, velocity is calculated as:

$$\omega = \frac{V_r - V_l}{W}\ (rad/s) \qquad (2.2)$$

Figure 2 shows how opposite rotation in motors with same speed enables the vehicle to rotate around its center.

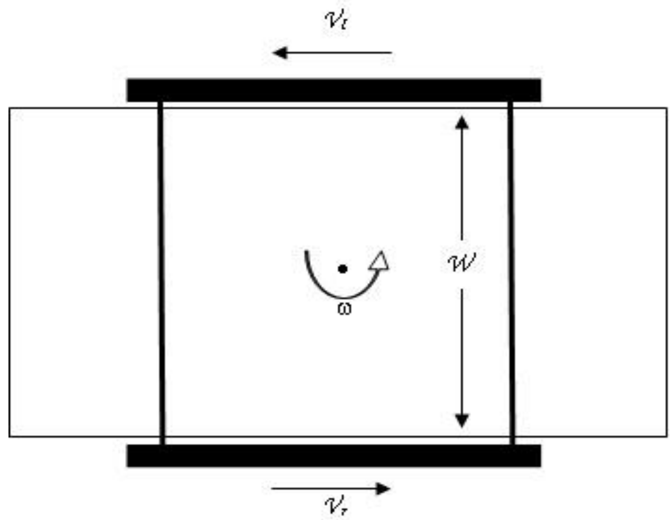


Figure 2: Rotation in Both Sides, Different Orientation, Same Speed

$$V_r = V_r = V \tag{2.3}$$

$$V = \frac{V_r + V_l}{2} = 0 \tag{2.4}$$

$$\omega = \frac{V_r - V_l}{W} = \frac{2V}{W} = \frac{V}{200} \tag{2.5}$$

Figure 3 shows no rotation in one motor and a normal rotation in the other motor, the vehicle turns whole at the center which is 370 mm away from the vehicle center.

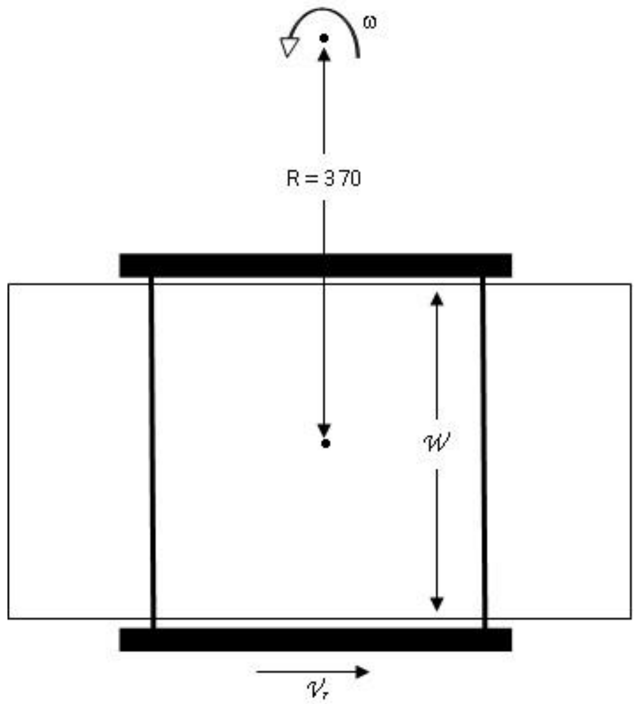


Figure 3: Rotation Just In one side

$$V_r = V \tag{2.6}$$

$$V_l = 0 \tag{2.7}$$

$$V = \frac{V_r + V_l}{2} = \frac{V}{2} \tag{2.8}$$

$$\omega = \frac{V_r - V_l}{W} = \frac{V}{W} = \frac{V}{400} \tag{2.9}$$

Figure 4 shows the same oriented rotation in the motors with various velocities: the vehicle turns whole of a center which can be from 370 mm away from the center to infinity due to various amount of variation of the speed.

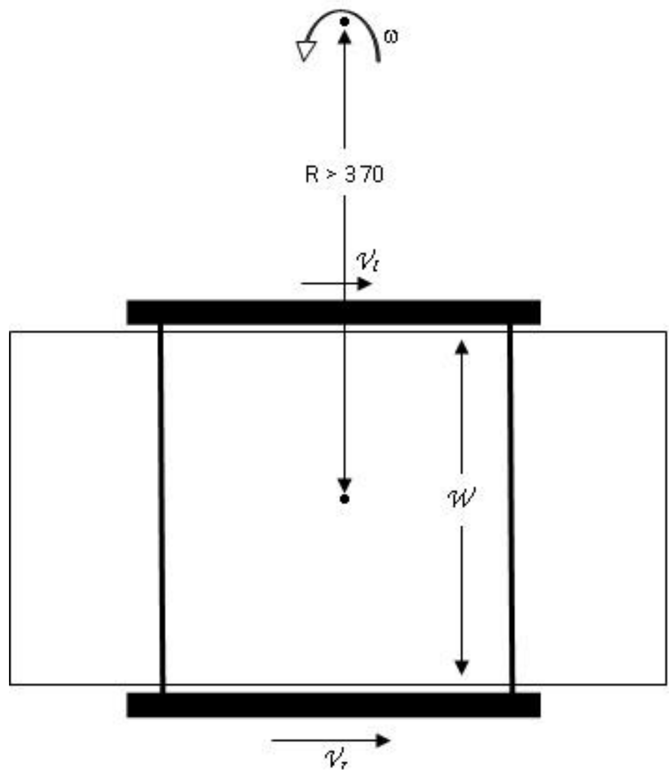


Figure 4: Rotation in Both sides, Same Orientation, Various Speeds

$$V_r = 2V \tag{2.10}$$

$$V_l = 2V \tag{2.11}$$

$$V = \frac{V_r + \ \ V_l}{2} = \frac{3}{2}V \tag{2.12}$$

$$\omega = \frac{V_r - \ \ V_l}{W} = \frac{V}{W} = \frac{V}{400} \tag{2.13}$$

Figure 5 shows the opposite rotation in motors with variation in speed; vehicle turns whole of a center which can be from the center of the vehicle to 370 mm away, due to various amount of variation of the speed.

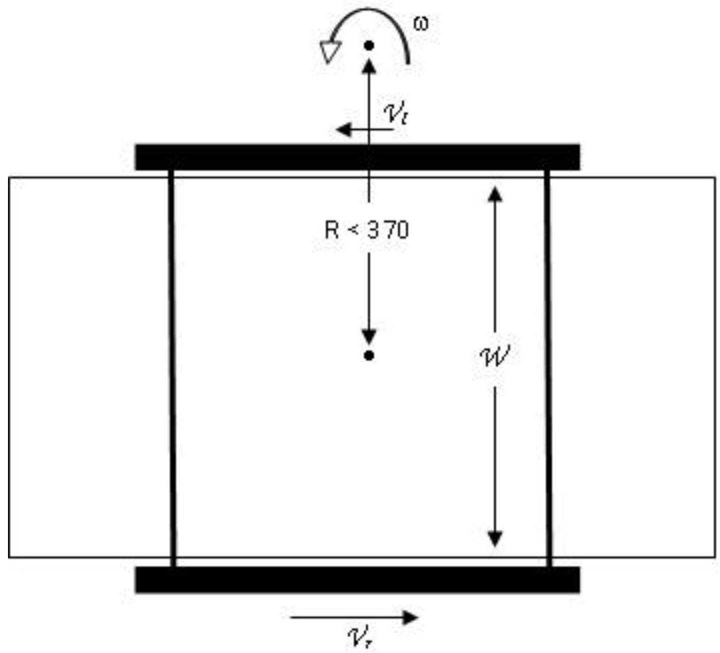


Figure 5: Rotation in Both Sides, Different Orientation, Various Speeds

$$V_r = 2V \tag{2.16}$$

$$V_l = -V \tag{2.17}$$

$$V = \frac{V_r + V_l}{2} = \frac{V}{2} \tag{2.18}$$

$$\omega = \frac{V_r - V_l}{W} = \frac{3V}{W} = \frac{3V}{400} \tag{2.19}$$

In addition, this variation of the speed is perfectly assigned with respect to the orientation of the turn. The big advantage of using this model is that it does not require too large an area for all types of the movements, turns and maneuvers.

3.1.4 Components of AGV

1. **Chassis**

- **Base and Supports:** The proposed AGV's chassis is made of a 3 millimeter thick rectangular sheet metal (400×700) millimeter at the bottom, supported by parallel and perfectly welded (20×40) millimeter profiles at the top in order to be able to stand the heavy weight of the other components and the maximum expected load. In addition the base is surrounded by the same perpendicular profile to hold the covers and withstand unexpected load, collisions, anything unpredicted or further from the project subject and other applications[1].
- **Two Motor Plates:** On each side there is special thick plate (4×70×240) millimeter to hold the motors. Each plate has 4 slots one for the motor shaft in the middle and the remaining 3 slots, 120 degree from each other, to tight and adjust the motor in the right place in order to make the belt sufficiently tight. This part is highly accurate therefore it is produced by the CNC milling.
- **A Platform for Lifter Motor:** There is one platform (8×85×85) millimeter, held by 4 legs wielded to the base for suspending the motor to an adequate position.
- **Electric Devices Stand:** Three stands are considered at the head of the AGV for kits and other electric devices to be hold.

---

[1] This vehicle has one other application which is 3-D environment scanner robot. For this application some other components, equipments and devices are assumed, designed, manufactured and assembled. For the scanner, a special holder is designed to be fixed to the head of the lifter so that it stays parallel to the ground all the time.
A specific control unit is assigned to control the movement in different directions and to run the scanner. This unit has a wireless system to receive commands from its manual controller and in addition the vehicle is equipped with a camera at the head for special situation. Direct eye contact and control is restricted and the projection is sent to the base via the wireless system so that the operator sees a wide view of the front side of the vehicle.

Figure 6 shows the chassis with all the components.

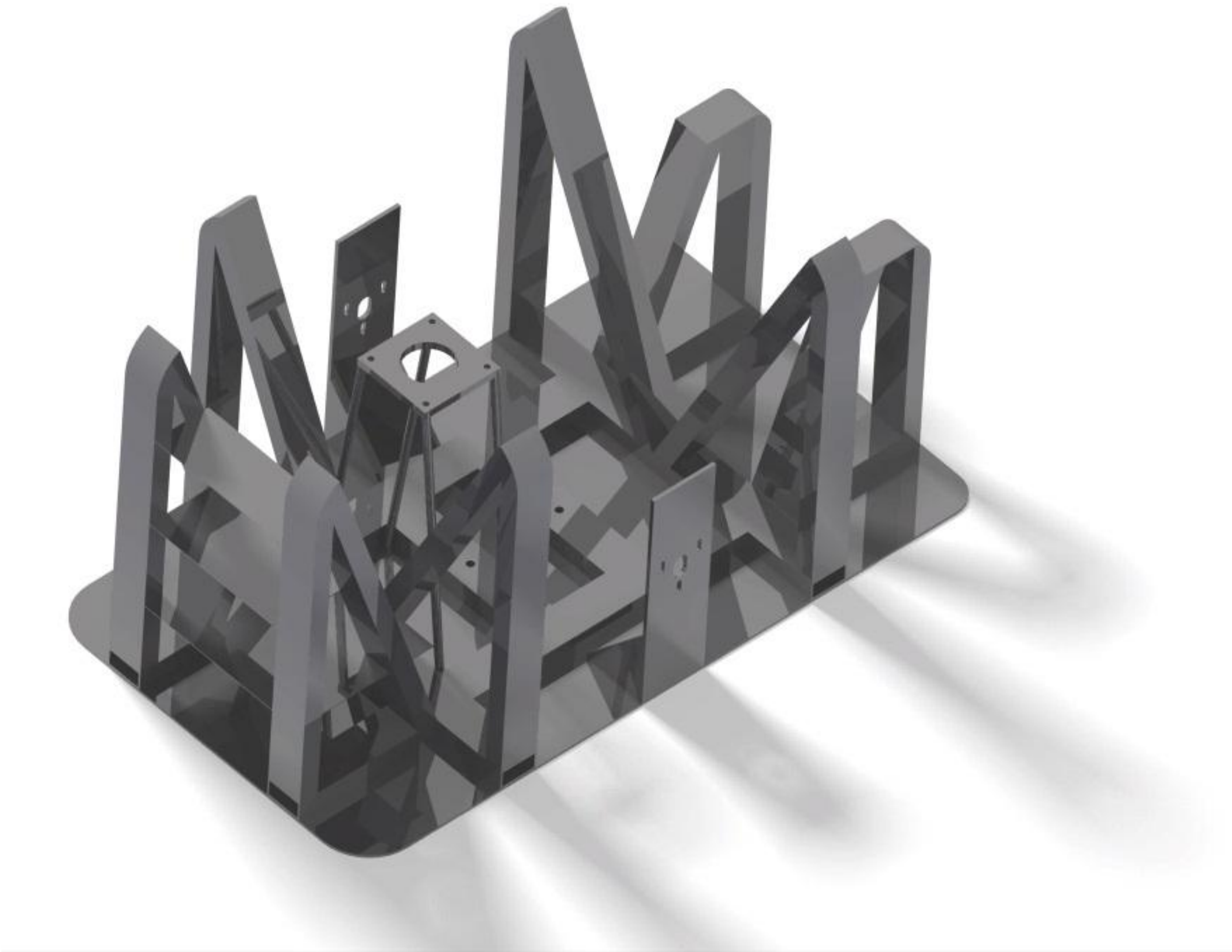


Figure 6: Chassis 3D Model

2. DC Motor Drivers

There are two 12V-3A DC electro motors adjusted and tighten to the *motor plates* on each side. These motors are considered to drive the AGV. Figure 7 shows the Motor Driver.

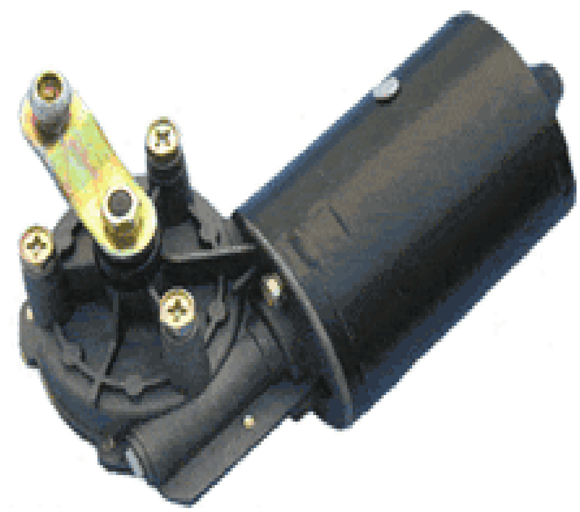


Figure 7: DC Motor Driver

3. **Power Transfer System:** (two sets, one for each side)

- **Driver Gear:** A spur gear is fixed to the motor output axis (D: 60 W: 35 millimeter). This gear is not only meant to transfer the power from the motor shaft to the driven gears, but also to keep the belt over the gear set in order to have continuous revolution under the load. This is done by special two fixture rings wielded to each side of this gear called gear guards.
- **Fixed Shafts:** At the bottom of the AGV there are two shafts (D: 15, L: 430 millimeter) parallel to the chassis with each 175 millimeter away from the center welded to the base. In order to be able to assemble the pinion gears on these shafts, the diameter is reduced using turning machining by 10 millimeter from

each head for 25millimeter. In this way the pinion gears are free to spin around the shaft with little friction.

- **Pinion Gears:** There are a pair of spur gears on each side (D: 120, W: 25 millimeter) seated on the fixed shaft with ball bearings. Consequently, these gears have only one rotational degree of freedom around the shaft.
- **Belts:** A 120 teeth timing belt is meant to transfer the power for each side.

Figure 8 shows the 3D model of Power Transfer System in 4 views.

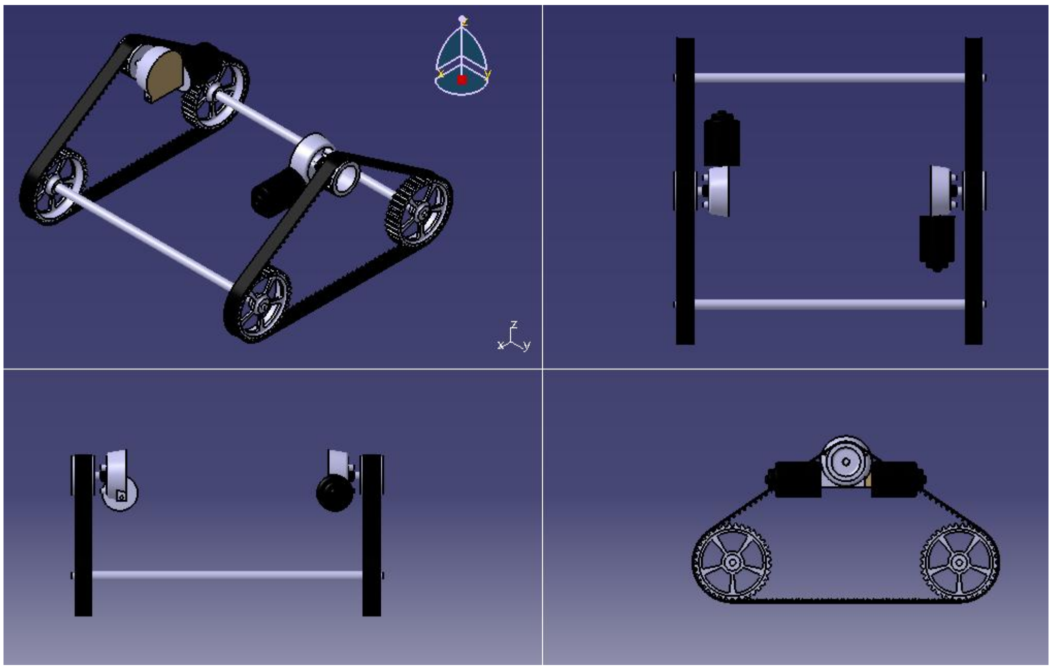


Figure 8: Power Transfer System 3D Model

4. Lifting System

- **Telescopic Lifting Device:** Due to probable varying heights of the stations, the AGV is predicted to be equipped to a lifting system in order to have this ability to adjust the height appropriate to each station. This lift consist of 3 telescopic threat shafts and a gear (D=150millmeter) involutes with another gear (D=70millimeter), driven by a motor.

- **Lifter Motor:** A12V-3A electro step motor is tighten to the pertinent platform to reach the height of the lifter gear.

Figure 9 shows the 3D model of Lifting System in four views.

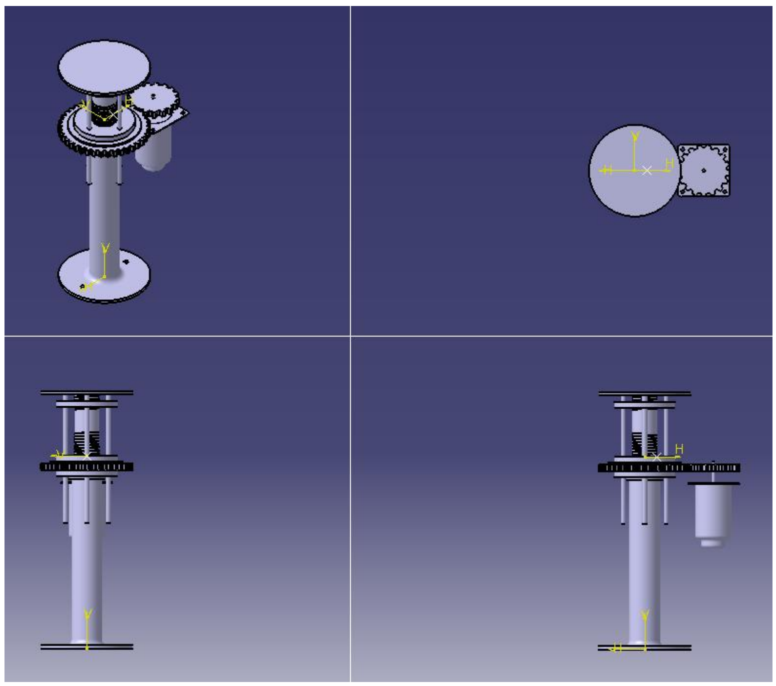


Figure 9: Lifting System 3D Model

5. Electrical components

- **Batteries:** Two 12V-4A batteries are assigned to supply the required energy for the drivers and the main unit. These batteries are placed at the back side of the vehicle to balance the weight and keep it close to the center.
- **Main Unit Kit:** This unit is the brain of the AGV. It has three 8Mb micro controllers which are meant to control all of the components for each of two

applications. It consists of three subsets, one processor for each, programmed with respect to the application requirements and expected respond. Figure 10 shows the picture of the main unit in which three micro controllers are seen.

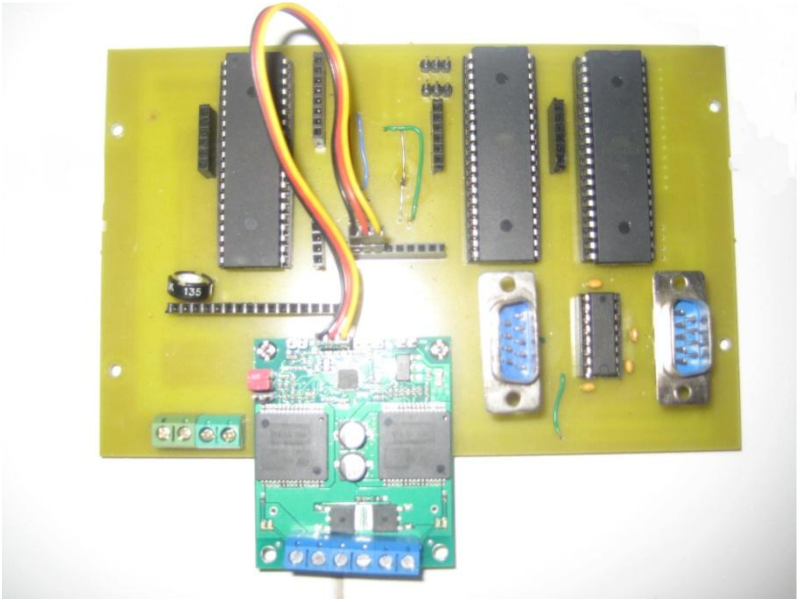


Figure 10: Main Unit Kit

I. **Application Controller Set:** This unit realizes the application mode (line follower or 3-D scanner), reads specific code sets and sends the commands to the pertinent subset to be processed. In addition this unit is connected to the communication unit which can receive and transmit commands and data with respect to its program. (the program is attached)

II. **Line Follower Set:** This unit is connected to the sensor set and reads them frequently and makes decisions in processing the received data. As mentioned earlier, this process is done in the assigned processor which is programmed appropriately to the line following AGV application. This program consists of three main parts: data source, decision making part and movement functions.(the program is available on Appendix A)

III. 3-D Scanner Set:

- **Sensors Board:** Five infrared sensors (Appendix B shows the operational functions) are placed in a kit at the bottom of the AGV. They are located 80 millimeter away from the center towards the head and 10 millimeter higher than the ground level. These sensors emit infrared light and measure the percentage of reflected as a bit. Consequently, the scientific range of this number is from 0 to 1024. Thus for the close object, this number tends to zero and for the far objects it tends to 1024. In this way the distance of the object to the sensor is detected if and only if the abject is shiny enough. As a result for matt objects which have less reflection however close they are the sensor detects a kind of far distance. On this board the sensors are placed one in the center, meant to be always just on the reflecting line, two sensors each 10millimeter away from the center to the sides, meant to be just after the line to report the deviation from the AGV from the line follower controller unit and the last two sensors each 40millimeter away from the center to the sides, meant to report junctions of the path. The figure 11 shows the sensor board layout, indicating the calling names and the sensors order.

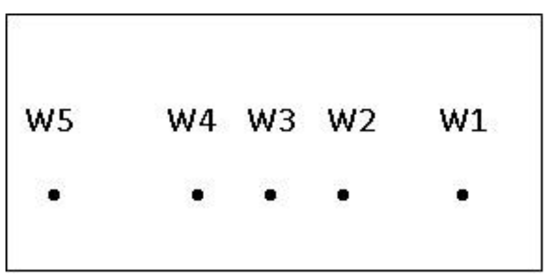


Figure 11: Sensor Board Layout

- **Monitor:** A 2 rows and 16 columns display gives the ability of reading data onboard. Figure 12 shows the picture of the Monitor.

Figure 12: Monitor

- **Motor Driver Kit:** This driver has power input from the batteries and output is divided into 127 units so that the speed is from 0 to 127. It also controls the polarization of outputs and provides the clockwise and counterclockwise rotation which leads to forward and backward linear movement. Considering the independent control for each side with respect to the program, the decision making unit or the 3-D scanner unit, the military movement model of the AGV is provided. (Appendix C shows the operational manual). Figure 13 shows the picture of Motor Driver Kit.

Figure 13: Motor Driver

- **Lifter Driver Kit:** This driver has power input from the batteries and the output is at constant speed with the ability of polarization. This driver kit responds to the received command from the main unit. Therefore the clockwise and counterclockwise rotation of the motor and the revolution of the sure pinion gear and the threat shaft of the lifter provide vertical movement of the lifter shaft up and down. Figure 14 shows the picture of Lifter Driver Kit.

Figure 14: Lifter Driver

- **Communication Unit:** Since this unit is meant to communicate with the host controller all the time for the both applications, it is considered as a part of the main unit, although this unit has no part in the making decisions. Figure 15 shows the picture of Wireless System. (Appendix D shows the operational manual).

Figure 15: Wireless System

- **Camera:** For the slave application, the 3D scanner, the vehicle is equipped with a camera at the head for special situations which direct eye contact and control is not provided or is possible but with some restriction. This camera provides a wide view of the front side of the vehicle and transmits the projection via the communication unit to the base (host controller). Therefore the operator is able to have a clear view and appropriate control of the vehicle. Figure 16 shows the picture of the Camera and its components.

Figure 16: Camera

- **Switch:** A switch is considered and placed at the back cover, connected to the electrical circuit of AGV after power supply. Therefore it turns the vehicle on and off.

6. **Covers**

- **Front Cover:** This part is designed for the front head of the vehicle. It is fixed to the chassis to cover and protect inner parts which are the most delicate and sensitive parts of the AGV. Figure 17 shows the 3D model of the Front Cover.

Figure 17: Front Cover 3D Model

- **Back Cover:** This part is designed for the back head of the AGV. It covers the batteries and holds the On/Off switch and the communication unit antenna. In addition, this part is screwed tighten to the chassis supports. Figure 18 shows the 3D model of the Back Cover.

Figure 18: Back Cover 3D Model

3.2 Path and Guide-Path Design

The essential capability of an AGV is the ability to transfer loads to distant locations through complex paths [45]. Moreover, the FMS has multiple station some in process which are not yet known as AGV targets, and some which awaiting loading/unloading known as AGV targets.

3.2.1 Interaction of Paths and Sensors

Each station has its own path; however this path might be partially in common with the other station's path. It is assumed that as soon as the AGV is dispatched to a station, the pertinent guide-path is selected and followed by the AGV. Meanwhile the sensor is collecting path data and sending them to the *line follower unit* to be processed and driving the motors according to the path at all times. Each guide-path might consist of three different types of turns and of course three different approaches to always catch up with the line:

I-Curve Turns: These types of turns are known as path deviation via deviation detector sensors on the sensor board and the guiding unit comes over them calling the deviation corresponding function (extended explanation in programming section). Figure 19 shows two pictures of Curve Turns.

Figure 19: Curve Turns

II-Intersections: These types of turns are known as junctions. They are perpendicular to the first path. The guiding unit is realized via the junction detector sensors on the sensor board and comes over them calling the "Turn Left" or "Turn right" function (extended explanation in programming section). Figure 20 shows two pictures of Intersections.

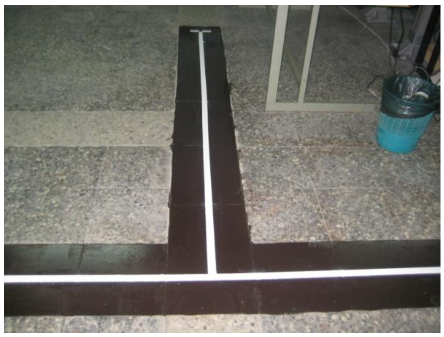


Figure 20: Intersections

III-U-Turns: These types of the turns happen at the end of path (deadlock) when the AGV has reached the target station and the guiding unit realizes that via all the sensors except W3 and comes over them calling U-Turn function (extended explanation in programming section). Figure 21 shows the picture of U-Turn sign.

Figure 21: U-Turn

3.2.2 Path's Specifications

As mentioned earlier (the interaction of the sensors and different colors) there are two colors chosen for the guide-path.

Basis Color: This color is a matt color with very low reflection drawn on the ground between stations and links them together. The basic width of this line is 300mm but it becomes greater when it gets closer to the junctions. This is to avoid failures happening while turnings. In this case the sensor board may go out of the basis path and read the data from the marble carpet of the shop floor which makes the plan unlikely and unpredictable.

Guiding Color: This color is a shiny color with high reflection drawn precisely at the middle of the basis path with 20 millimeter width to keep the deviation detector sensors exactly on its side edges. It goes between stations and ends up in a specific form. All the paths end up with a special T-shape without the head to provide a special opportunity for the guiding unit to realize the deadlocks which are meant to be the stations. Figure 22 shows the guide path layout and the position of stations.

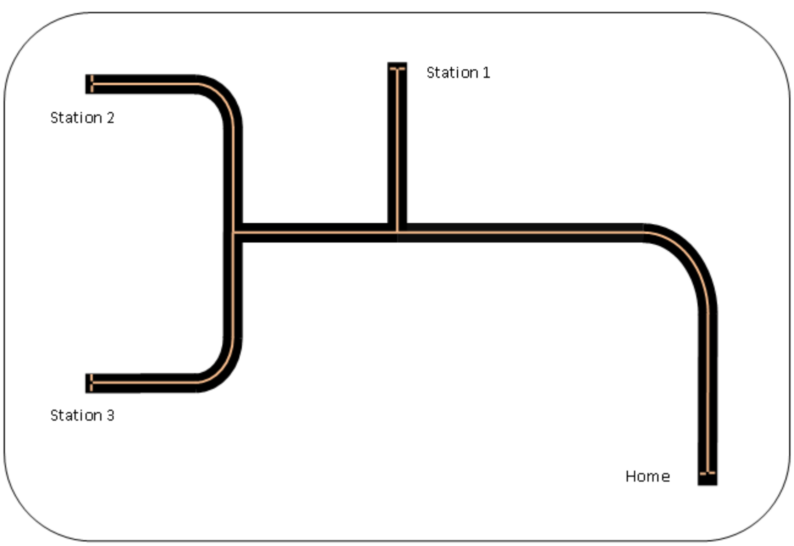


Figure 22: Guide Path Layout

3.3 Work Stations Information

Station information basically includes the height of the station and the path addressed for each station to the others. The information about each station is stored in the data base of the host controller unit. Before the AGV is dispatched to a station, this information is loaded to the guiding unit so that the AGV adjusts the turning priorities as well as the lifter height adequate to the height of the station. Moreover, for the next command the guiding unit uses this updated data as a reference position.

3.4 AGV Scheduling

The aim of AGV scheduling is to dispatch the AGV to achieve the goals for a batch of traveling tasks under certain conditions such as minimum lead-time and maximum reasonable speed within minimum deviation. The goals are normally related to the processing times, utilizing data resources and minimizing the AGVs traveling distance which leads to minimum total travel time. [46]

A heuristic scheduling algorithm finds the end point of the late task assigns a path and starts a point of the next task; a list of probable tasks is predefined. Each task (either loading or unloading) contains three operations.

3.4.1 Idling

When the AGV turns on, it goes on idle mode waiting for the host computer command. This command might be one or more numbers implying different priorities of the possible path. Considering four stations (three stations and one home), there are always three possibilities for each station. Therefore, there are generally twelve specific routs between them. Each of these routes is given a code and defined by priorities in junctions to the right or left.

3.4.2 Moving

As soon as the AGV gets the command, it starts to move on the line with the updated priorities to reach the destination station.

3.4.3 Stopping

When the AGV reaches the destination, it does a U-turn and turns backs on the line ready, on the idle mode, waiting for the next command.

3.5 Software Design

3.5.1 Algorithm

The success of the AGV system is highly dependent on the quality of the logic control design. In other word, in order to operate an AGV system efficiently, AGVs require an extensive controller system.

The responsibilities of the AGV controller include the following decisions [47]:

- How to assign idle position till the picking up of a request? This is referred to as the dispatching problem.
- Which route is to be taken to the station and from the station to the next destination? This is referred to the routing problem.
- How to employ resources, guide paths and functions to avoid system deadlock while traveling? This is referred to as the movement control problem.

The next chapter discusses the challenges of applying an adequate algorithm and the algorithm layout.

3.5.2 Programming

The next step after algorithm is to write the program according to the algorithm. Due to employed processors, the programming language has to be Basic. The program is written in three phases (Appendix A):

- Inputs
- Guiding section
- Functions

3.5.3 Interface Commander

To simulate an integrated manufacturing shop floor, an Interface Commander is designed to play the role of the host computer. In real integrated FMS, it is the host computer which sends the commands to the AGV controlling unit with respect to the requirements. Since the integration is out of the subject of this thesis, this commander is designed to send the commands in the absence of the host computer. This interface not only sends the commands to the AGV but also receives all the AGV information and data all the time such as motors velocity, deviation, and AGV position. This data is used in chapter five to show the process of AGV movements and deviation corrections.

3.6 Battery Management

Battery management of the AGV is an issue that is not being strongly focus on but as an electrical vehicle is considered, battery State Of Charge (SOC) estimation becomes an increasingly important issue in terms of both extending the lifetime of the battery and displaying the usable charge to the user before recharging. However, the SOC cannot be measured directly, but rather must be estimated based on measurable battery parameters such as voltage and current.

Figure 23 shows picture of AGV while starting operation from Home position.

Figure 23: AGV Starting Operation

Figure 24 shows picture of AGV while performing a curve turn to the left.

Figure 24: AGV on a curve turn to the left

Figure 25 shows the AGV remaining on the line after performing a curve turn to the left.

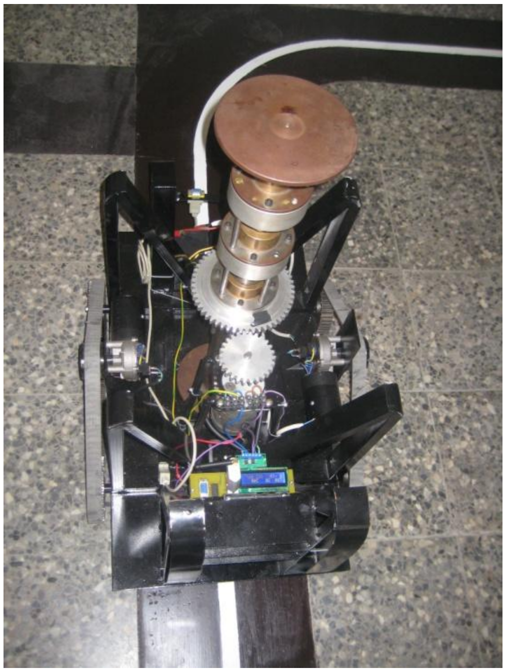


Figure 25: AGV after performing a curve turn to the left

Figure 26 shows the AGV facing two intersections when a decision is needed to be made.

Figure 26: AGV having two intersections in front

Figure 27 shows the AGV is performing a 90 degree turn to the left while remaining on the line.

Figure 27: AGV performing 90 degree turn to the left

Figure 28 shows the AGV is performing a curve turn to the right while it remains on the line.

Figure 28: AGV performing curve turn to the right

Figure 29 shows the picture of AGV just performed a curve turn heading to a station.

Figure 29: AGV heading to a station

Chapter 4

METHODOLOGY (STEERING METHOD)

Steering systems perform several roles such as idling, dispatching, deviation and station recognition. In addition, a comprehensive algorithm must be written that can efficiently accommodate and manage the different roles to avoid conflict.

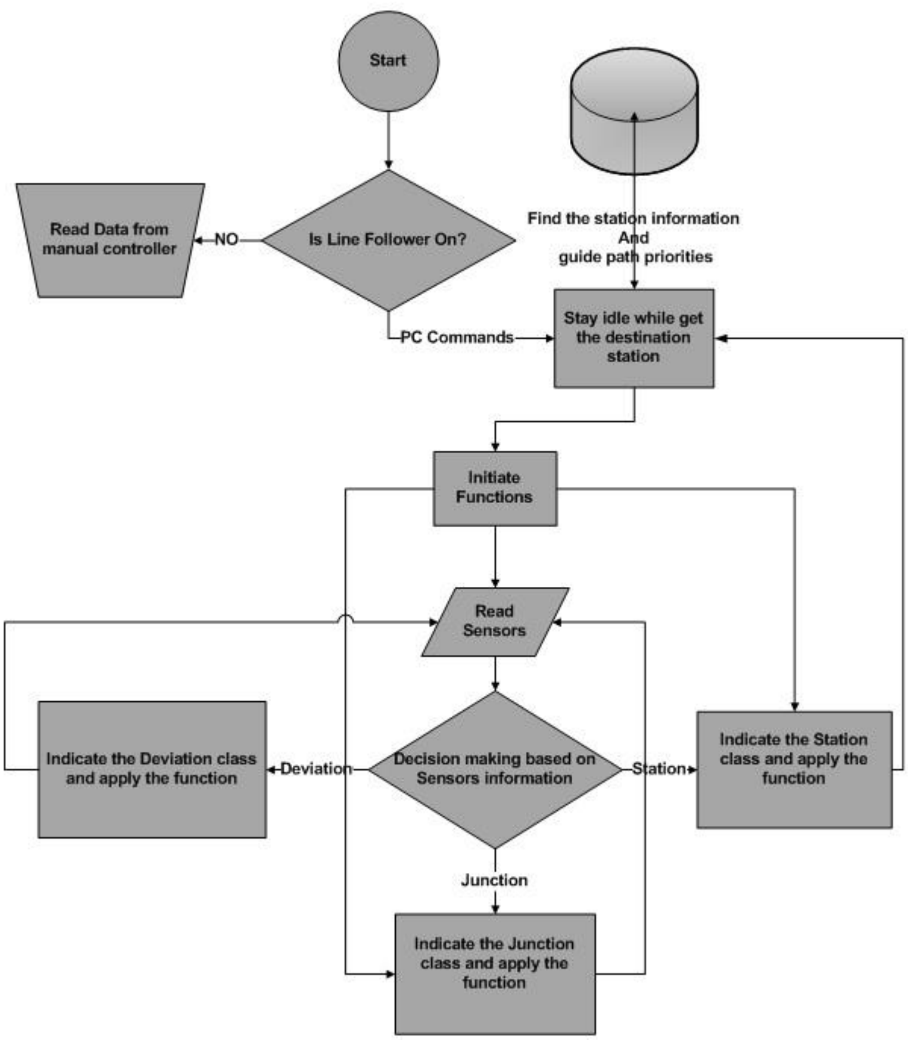


Figure 30: Algorithm

4.1 Algorithm

To illustrate the steering system sequences the algorithm is drawn in the figure 30.

4.1.1 Mode Selection

As earlier mentioned, in the decision making unit and also as it is seen from the algorithm, for the steering system to jump on the line follower mode (AGV), there is special code sent by the host controller.

4.1.2 Initiating the Functions in Idling Mode

This step is done just after the mode selection. In this step AGV remain in idle position waiting for constant and variable values to be read and set. In this position the steering system does not dispatch the AGV without a certain path to a certain destination. It keeps the AGV idle on the line, ready to get the two digit code dedicating on a certain path to a certain station. In this system every station's path to the others is known by two digit code. Considering, four stations and three possible destinations for each, there are twelve paths between the four existing stations which are known by a special code. All of these 12 paths act like an address to the AGV giving the priorities of turn (like addressing a vehicle in a city with several streets). In addition, these addresses plus the height information of each station is stored in a data base. For instance, for the AGV to go from home to the first station, considering the path layout in figure 22, the AGV needs to turn the first junction to the right. Like wisely, from station 1 to station 3 The AGV needs to turn the first junction to the right and second junction to the left.

When all the needed information for dispatching is given, the steering system changes the idle mode to dispatch mode. As a result the AGV moves forward and follows the line to the destination station. Meanwhile, the guiding system is checking and controlling three possibilities all the time.

4.1.3 Deviation

Deviation is defined by the differentiation of W2 and W4 (deviation control sensors from sensors board, chapter 3 sensors board).

$$\text{Deviation} = W4 - W2 \tag{4.1}$$

(Positive value represents deviation to the left and negative value represents deviation to the right)

For the AGV not to deviate from the guideline and to be at a reasonable distance away (deviation) from it, the deviation range is classified in to eight levels. Each deviation is treated according to the class of the deviation. The supervisory control resets the velocity of left and right motor according to the class and orientation of deviation.

Table 1 shows the classification of deviation and velocity difference.

Table 1: Deviation Classification

Class	**Deviation Range** (Bytes)	**Velocity Difference** (mm/s)
1st	0< Deviation<200	0
2nd	200< Deviation<300	10
3rd	300< Deviation<400	20
4th	400< Deviation<500	30
5th	500< Deviation<600	40
6th	600< Deviation<700	50
7th	700< Deviation<800	60
8th	800< Deviation<1024	70

If deviation is to the right, the Velocity Differentiation is applied directly and if deviation is to the left, the Velocity Differentiation is applied negatively. In the end, Velocity Difference affects the initial vehicle speed by a simple formula as shown below:

$V_r = V + \text{Velocity Difference}$ (4.2)

$V_l = V - \text{Velocity Difference}$ (4.3)

However, there are two consequences that must be taken to the account which may change the simplicity of this formula. As mentioned in chapter three motor drivers speed is divided in to 127 units (from 0 to 127). It means that as earlier shown, the formula finds out any value out of range which is not only unacceptable for the motor driver but also may cause a collapse condition. Therefore, for any value less than zero, the direction changes from forward to backward with the absolute value and for any velocity more than 127(mm/s), the velocity remains 127(mm/s) and the amount of exceeded velocity differentiates from the opposite side to maintain the balance. Table 2 shows some examples of different deviation class.

Table 2: Deviation Samples

Deviation class indicator	Initial Velocity V(mm/s)	Velocity Difference (mm/s)	V_r (mm/s) (V+Velocity Difference)	V_l (mm/s) (V-Velocity Difference)
1st	70	0	70 forward	70 forward
4th	70	30	100 forward	40 forward
7th	70	60	127 forward	3 forward
8th	70	70	127 forward	13 backward

Figure 31 illustrates some of the possible deviations of sensor board from the guide line.

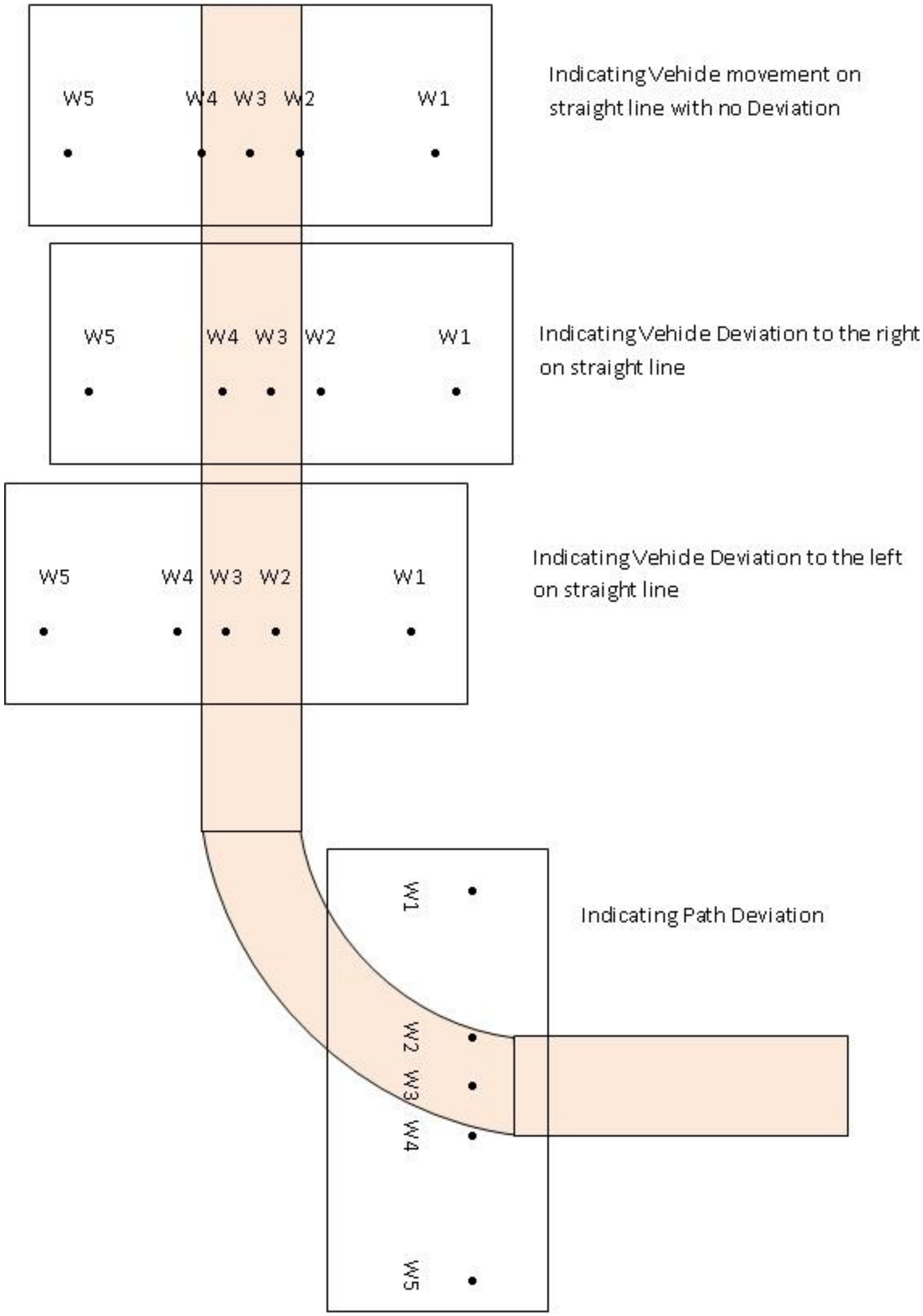


Figure 31: Possible Deviations

This procedure guarantees the movement of the vehicle on the straight guide line as well as the curve guide line. Due to supervisory control corrections, on the straight guide line deviation does not exceed the first class. However, in two possible situations the deviation class may exceed the first level. The first possibility happens when the AGV starts moving on the line and it is not yet precisely on the guide line. The second possibility is when the AGV goes on the curve guide line in which it is highly possible to reach the ultimate deviation class.

The relevant empirical result mathematically and graphically will be shown in chapter 5

4.1.4 Junction Detection

Junctions are detected by W1 and W5 (junction control sensors from the sensors board, chapter three sensors board) and the number of the detected junctions increases by one if they detect a junction. Whenever the number of detected junction matches the station address, the supervisory system turns the AGV 90 degree to the head of the station.

4.1.5 Station Detection

Whenever all the sensors except W3 (W1, W2, W4 and W5) see the line, it means for the supervisory system that the AGV has reached the destination station. Therefore, it performs a U-turn and stops on the line again and the mode changes to the idle and it again waits for another dispatching command.

4.2 Program

Due to processors type the Basic programming language is employed. It is consisted of three phases.

In order to understand the sequence of the programming, it is provided in appendix A with extra explanation.

4.2.1 Inputs

Inputs are divided in to two general types

- Constants:

To initialize the AGV position, some constant values must be defined. For example for the first mission, motors are set on forward mode. The constant values are:

Initial position: Home

Right motor: Forward

Left motor: Forward

Detected Junction: 0

- Variables:

For the Technicians to change the velocity of the AGV without changing all of related values the whole program is written parametrically respect to the V-value, dedicating to the velocity and also certain number of variables are defined for the AGV to go to different stations, dedicating to priorities of turns.

4.2.2 Guiding Section

This section is in charge of supervising the AGV situation all the time and calling the adequate function for each facing situation.

4.2.3 Functions

In this phase all of the needed functions are defined to be called to guide the AGV appropriate to the AGV situations.

Chapter 5

OPERATION RESULTS

Due to this fact that all of the AGV components are made by hand as well as the justifications and settings, the straight movement of AGV might have a functional error deviation from the straight line. However, the combination of the error deviation and the guide line deviation are always controlled and covered by the supervisory unit.

In order to see how the supervisory treats, controls and over comes the error deviation angle and path deviation, the AGV is examined in two situations and four correlating variables (Real Deviation, Classified Deviation, V_l, V_r) in each loop of the program are transmitted to the host controller of the AGV and issued on a list.

5.1 Straight Line Examination

This examination shows how the AGV finds the path and its own precise position on the straight line. Table 3 shows first examination results.

Table 3: Straight Line Results

<table>
<tr><th>Real Deviation (Bytes)</th><th>Velocity Difference (mm/s)</th><th>V_l (mm/s)</th><th>V_r (mm/s)</th><th>Section & Explanation</th></tr>
<tr><td>-568</td><td>-40</td><td>30</td><td>110</td><td rowspan="24">1
In this section as it is seen the Real Deviation is a negative value therefore it is classified according to negative and has a direct effect on the velocity of the motors. As it was supposed this value is added to the initial velocity for the left motor and differentiates for the right one.</td></tr>
<tr><td>-471</td><td>-30</td><td>40</td><td>100</td></tr>
<tr><td>-485</td><td>-30</td><td>40</td><td>100</td></tr>
<tr><td>-539</td><td>-40</td><td>30</td><td>110</td></tr>
<tr><td>-534</td><td>-40</td><td>30</td><td>110</td></tr>
<tr><td>-586</td><td>-40</td><td>30</td><td>110</td></tr>
<tr><td>-610</td><td>-50</td><td>20</td><td>120</td></tr>
<tr><td>-622</td><td>-50</td><td>20</td><td>120</td></tr>
<tr><td>-639</td><td>-50</td><td>20</td><td>120</td></tr>
<tr><td>-679</td><td>-50</td><td>20</td><td>120</td></tr>
<tr><td>-729</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-732</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-749</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-752</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-777</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-759</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-776</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-767</td><td>-60</td><td>100</td><td>127</td></tr>
<tr><td>-654</td><td>-50</td><td>20</td><td>120</td></tr>
<tr><td>-497</td><td>-30</td><td>40</td><td>100</td></tr>
<tr><td>-342</td><td>-20</td><td>50</td><td>90</td></tr>
<tr><td>-172</td><td>0</td><td>70</td><td>70</td></tr>
<tr><td>-86</td><td>0</td><td>70</td><td>70</td></tr>
<tr><td>-14</td><td>0</td><td>70</td><td>70</td></tr>
<tr><td>82</td><td>0</td><td>70</td><td>70</td><td rowspan="3">2
In this section as it is seen the</td></tr>
<tr><td>136</td><td>0</td><td>70</td><td>70</td></tr>
<tr><td>210</td><td>10</td><td>80</td><td>60</td></tr>
</table>

210	10	80	60	Real Deviation is a positive value therefore it is classified according to positive and direct effects of the velocity of the motors. As it was supposed this value is added to the initial velocity for the left motor and differentiates for the right one.
339	20	90	50	
309	20	90	50	
337	20	90	50	
346	20	90	50	
383	20	90	50	
457	30	100	40	
484	30	100	40	
524	40	110	30	
439	30	100	40	
192	0	70	70	
232	10	80	60	
222	10	80	60	
197	0	70	70	
172	0	70	70	
154	0	70	70	
78	0	70	70	
124	0	70	70	
84	0	70	70	
19	0	70	70	

To see more illustrative view of the result the following graphs have been drawn using the straight line examination data.

Figure 32 shows the deviation classification graph.

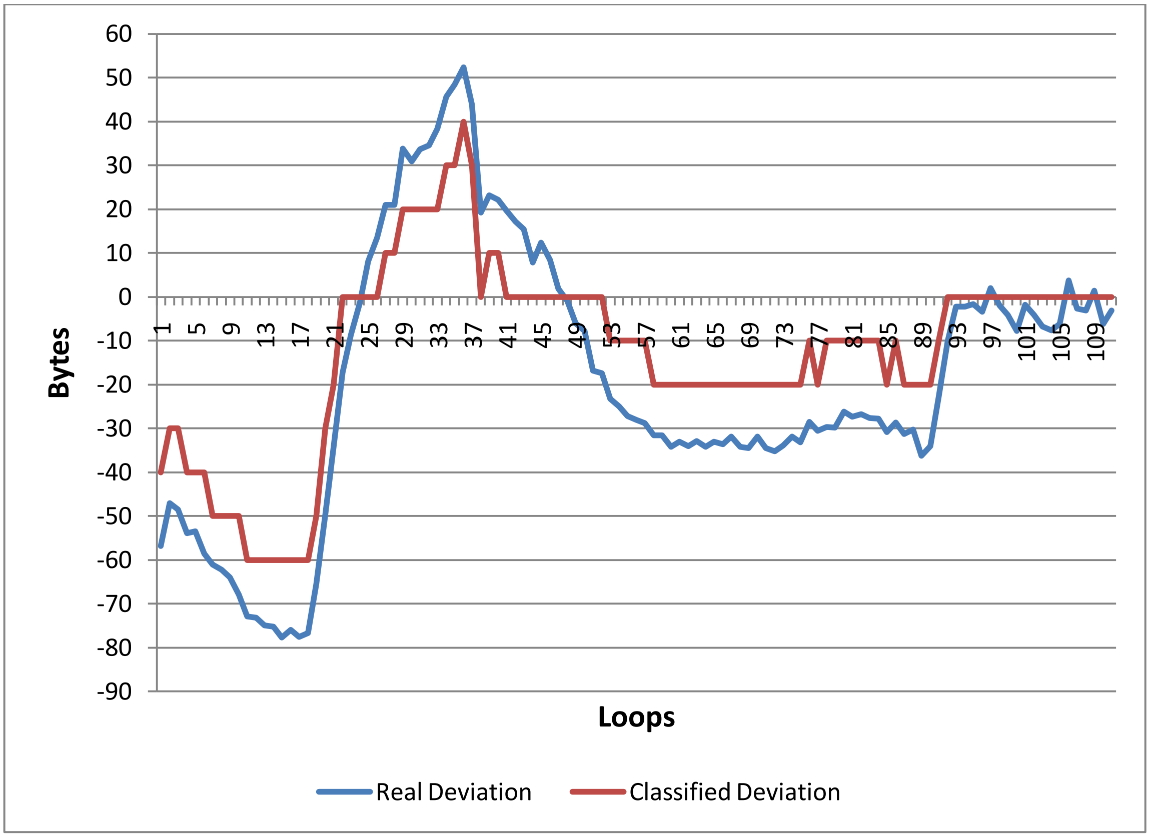


Figure 32: Deviation Classification

It is seen that the classification graph dies out approximately after 100 loops. It shows the needed loop time for the AGV to find precise position on a straight guide line. Moreover, it is seen that after the 100th loop, the deviation is almost to the left which implies error deviation of the vehicle.

In figure 33, the graph shows the velocity variation of the left and right motor.

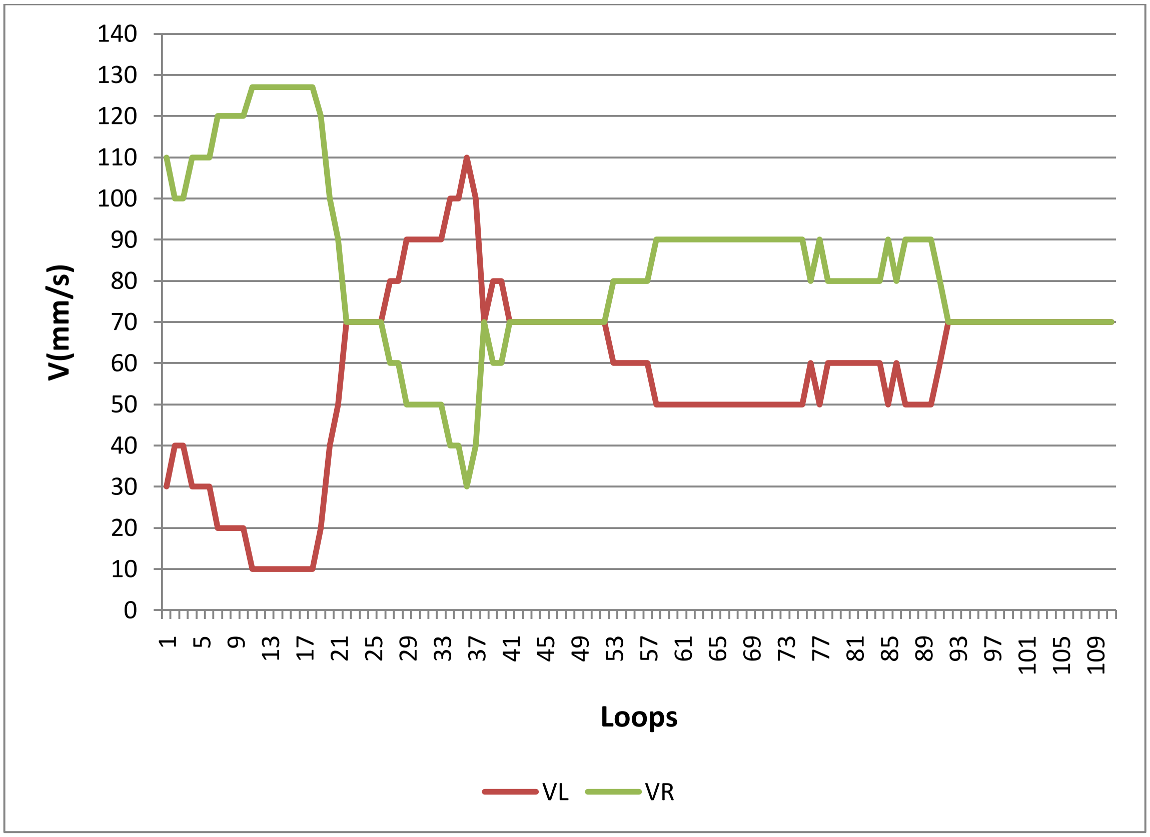


Figure 33: Velocity Variation of the Left and Right Motor on Straight Line

It is seen approximately after 100 loops the velocity variation of the left and right motors tend to 70 which is the specified speed for the AGV.

5.2 Curve Line Examination

This examination shows how the AGV finds the path and its precise position on the straight line.

Table 4 shows curve line examination result.

Table 4: Curve Line Result

Real Deviation (Bytes)	**Velocity Difference** (mm/s)	**V_l** (mm/s)	V_r (mm/s)
197	0	70	70
172	0	70	70
154	0	70	70
78	0	70	70
124	0	70	70
84	0	70	70
19	0	70	70
-9	0	70	70
-60	0	70	70
-77	0	70	70
-168	0	70	70
-174	0	70	70
-172	0	70	70
-86	0	70	70
-14	0	70	70
82	0	70	70
136	0	70	70
210	10	80	60
221	10	80	60
250	10	80	60
280	10	80	60
339	20	90	50
309	20	90	50
337	20	90	50
346	20	90	50
383	20	90	50
370	20	90	50
390	20	90	50
420	30	100	40
476	30	100	40
492	30	100	40
498	30	100	40
524	40	110	30
540	40	110	30
587	40	110	30
612	50	120	20
658	50	120	20
709	60	127	7
746	60	127	7

805	70	127	-13
820	70	127	-13
812	70	127	-13
801	70	127	-13
750	60	127	7
714	60	127	7
668	50	120	20
632	50	120	20
580	40	110	30
550	40	110	30
530	40	110	30
480	30	100	40
484	30	100	40
460	30	100	40
445	30	100	40
373	20	90	50
387	20	90	50
370	20	90	50
346	20	90	50
339	20	90	50
337	20	90	50
309	20	90	50
280	10	80	60
250	10	80	60
221	10	80	60
210	10	80	60
136	0	70	70
82	0	70	70
-14	0	70	70
-86	0	70	70
-60	0	70	70
-17	0	70	70

To see more illustrative view of curve line examination result the following graphs have been drawn.

Figure 34 shows the deviation classification graph.

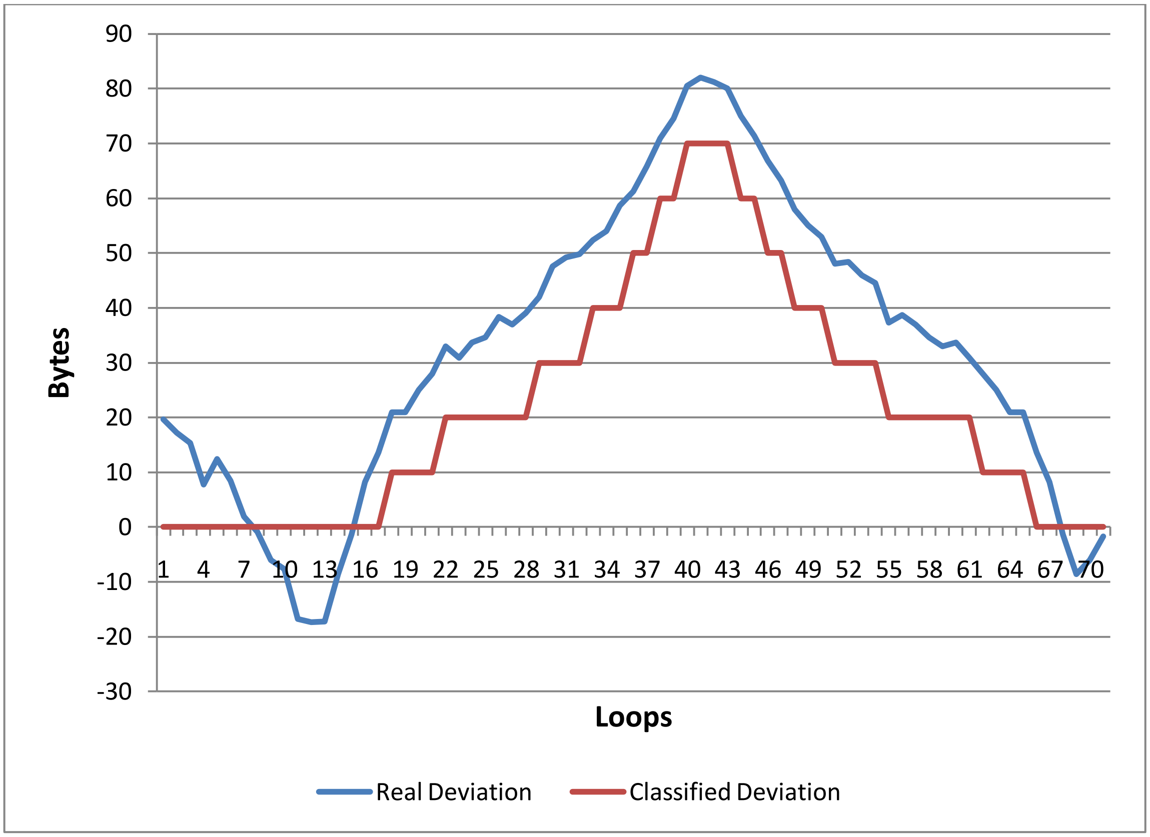


Figure 34: Deviation Classification

It is understood from the classified deviation that after the vehicle finds the line deviation remains in first class till path deviation occurs. While path has been divided, the deviation class is high. When the deviation is getting over, the deviation graph dies out again. This graph perfectly shows how the AGV comes over the path deviations (happens in curve line) and remains on the line.

In figure 35, the graph shows the velocity variation of the left and right motor on curve line.

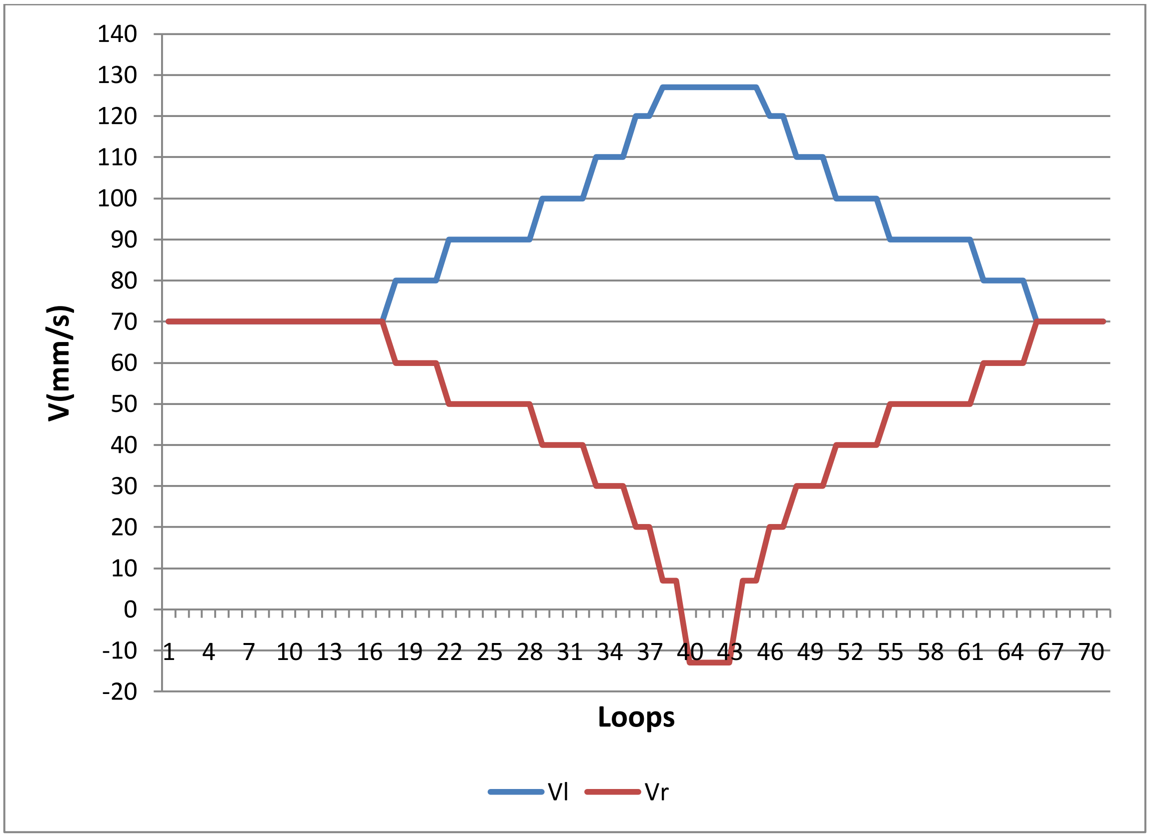


Figure 35: Velocity Variation of Left and Right Motor on Curve Line

This graph shows the normal speed of the AGV (both side same speed) before path deviation. When the AGV reaches the curve path, it turns respect to the turn changing speed balance of left and right.

Chapter 6

CONCLUSION AND FUTURE WORKS

An Automated guided vehicle (AGV) is defined as a set of cooperative driverless vehicle, which is used on manufacturing floor and coordinated by a centralized or distributed computer-based control system. The main usage of them as mentioned is to facilitate automation process of doing manufacturing subjects. In this practical research, according to the instructions of earlier study, an AGV have been made. Moreover a guide line has been provided with the mentioned specifications as shown in chapter three.

The primary goal of the AGV was to travel between stations. Therefore, the designed AGV has been examined numerously between all of the stations. Observations proved in every part of the testing procedure the AGV was able to get the commands, follow the line, find the appropriate rout, recognize the station, stop, and report its position.

Figures (23-29) show some pictures of the AGV while operating on different situations from home position to the station three.

Secondary goal of this thesis was to increase flexibility of the AGV. This is also successfully provided using the simple instructions of path designing. The guide path specifications are shown in figures (19-21).

It is mentioned previously that the battery management is not an issue being strongly focus on in this thesis. In this respect Battery State of Charge (SOC) estimation has become an increasingly important issue in terms of both extending the lifetime of the battery and displaying the usable charge to the user before recharging of the AGV. I believe it can be taken to improve the performance and efficiency of AGV. This system can measure the needed energy to operate and finish a task and automatically compare it with its current battery charge. If there is not enough energy to finish the task then recharging will be proceed before the ordered task. The SOC cannot be measured directly, but rather must be estimated based on measurable battery parameters such as voltage and current which is offered as future work.

The main focus of this study was to use a single AGV for all of the required material handling. Considering the recharging issues and hiring multiple AGVs in a same time in a manufacturing fool with several pallets makes the plan much more complex. In such plan the proposed logic is no longer applicable due to the nature of the employed software and hardware. In this case replacing the micro controllers by a Programmable Logic Controller (PLC) with higher accuracy, reliability and adaptively to other machines seems to be necessary as well as using a fuzzy logic appropriate to the new plan.

REFERENCES

[1] Ravazzi, P., Villa, A., *Economic Aspects of Automation*. (2009). Springer Handbook of Automatin, Part A, PP. 93-116

[2] Chadwick-Jones, J. K., *Automation behaviour: A social psychological*, (1969), study'Wiley-Interscience, (London and New York), (ISBN0471143006) xi, PP. 168.

[3] Basnet, C., Mize, J.H. *Scheduling and control of manufacturing systems: a critical review*, (1994), International Journal of Computer Integrated Manufacturing, Vol.7, PP. 340±355.

[4] Rachamadugu, R., Stecke, K.E., *Classification and review of FMS scheduling procedures*, (1994), Production Planning and Control, Vol.5, PP. 2±20.

[5] Han, M.H., McGinnis, L.F. *Control of material handling transporters in automated manufacturing*, (1989), IIE Transactions, Vol.21, PP. 184±190.

[6] Schriber, T.J., Stecke, K.E., *Machine utilization achieved using balanced FMS*, (1988), production ratios in a simulated Setting, Annals of Operations Research, Vol.15, PP. 229±267.

[7] Sabuncuoglu, I., Hommertzheim, D.L., *Dynamic dispatching algorithm for scheduling machines and AGVs in a flexible manufacturing system*, (1992), International Journal of Production Research, Vol.30, PP. 1059±1080.

[8] Sabuncuoglu, I., Hommertzheim, D.L., *Experimental investigation of FMS due-date scheduling problem: evaluation of machine and AGV scheduling rules*,

(1993) International Journal of Flexible Manufacturing Systems, Vol.5, PP. 301±324.

[9] Wilhelm, W.E., Shin, H.M., *Effectiveness of alternate operations in a flexible manufacturing system*, (1985), International Journal of Production Research, Vol.23, PP. 65±79.

[10] Chen, I.J., Chung, C.H., *Effects of loading and routing decisions on the performance of flexible manufacturing systems*, (1991), International Journal of Production Research, Vol.29, PP. 2209±2225.

[11] Khoshnevis, B.H., Chen, Q.M., *Integration of process planning and scheduling functions*, (1991), Journal of Intelligent Manufacturing, Vol.1, PP. 165±176.

[12] Rachamadugu, R., Nandkeolyar, U., Schriber, T.J., *Scheduling with sequence flexibility*, (1993), Decision Science, Vol.24, PP. 315±241.

[13] Lin, G.Y., Solberg, J.J., *Effectiveness of routing control*, (1991), The International Journal of Flexible Manufacturing Systems, Vol.3, PP. 189±211.

[14] Chandra, P., Li, S., Stan, M. *Jobs and tool sequencing in an automated manufacturing environment*, (1993), International Journal of Production Research, Vol.31, PP. 2911±2925.

[15] Kim, K.H., Jeon S.M., Ryu, K.R., *Deadlock prevention for automated guided vehicles in automated container terminals*, (2006), OR Spectrum, Vol.28, pp. 659-679.

[16] Maxwell, W.L., Muckstadt, J.A., *Design of Automatic Guided Vehicle Systems*, (1982), IIE Transactions, Vol. 14, No. 2, pp. 114–124.

[17] Gaskin, R.J., Tanchoco, J.M., *Flow Path Design for Automated Guided Vehicle Systems*, (1987), International Journal of Production Research, Vol. 25, No. 5, pp. 667–676.

[18] Goetz, W.G., Jr., Egbelu, P.J., *Guide Path Design and Location of Load Pickup/Drop-off Points for an Automated Guided Vehicle System*, (1990), International Journal of Production Research, Vol. 28, No. 5, pp. 927–941.

[19] Sun, X.C., Tchernev, N., *Impact of Empty Vehicle Flow on Optimal Flow Path Design for Undirectional AGV Systems*, (1996) International Journal of Production Research, Vol. 34, No. 10, pp. 2827–2852.

[20] Afentakis, P., *A Loop Layout Design Problem for Flexible Manufacturing Systems*, (1989) International Journal of Flexible Manufacturing Systems, Vol. 1, No. 2, pp. 175–196.

[21] Bartholdi, J.J., Platzman, L.K., *Decentralized Control of Automatic Guided Vehicles on a Simple Loop*, (1989), IIE Transactions, Vol. 21, No. 1, pp. 76–81.

[22] Sharp, G.P., Liu, F.-H.F., *An Analytical Method for Configuring Fixed-Path, Closed-Loop Material Handling Systems*, (1990), International Journal of Production Research, Vol. 28, No. 4, pp. 757–783

[23] Kouvelis, P., Kim, M., *Unidirectional Loop Network Layout Problem in Automated Manufacturing Systems*, (1992), Operations Research, Vol. 40, No. 3, pp. 533–550.

[24] Egbelu, P.J., *Positioning of Automated Guided Vehicles in Loop Layout to Improve Response Time*, (1993), European Journal of Operational Research, Vol. 71, No. 1, pp. 32–44.

[25] Banerjee, P., Zhou, Y., *Facilities Layout Design Optimization with Single Loop Material Flow Path Configuration*, (1995), International Journal of Production Research, Vol. 33, No. 1, pp. 183–203.

[26] Chang, S.H., Egbelu, P.J., *Dynamic Positioning of AGVs in a Loop Layout to Minimize Mean System Response Time*, (1996), International Journal of Production Research, Vol. 34, No. 6, pp. 1655–1674.

[27] Bozer, Y.A., Srinivasan, M.M., *Tandem Configurations for Automated Guided Vehicle Systems Offer Simplicity and Flexibility*, (1989), Industrial Engineering, Vol. 21, No. 2, pp. 23–27.

[28] Kim, C.W., Tanchoco, J.M.A., *Conflict-free shortest-time bidirectional AGV routing*, (1991), Int. J. Production Research, Vol. 29, pp. 2377-2391.

[29] Kim, C.W., Tanchoco, J.M.A., *Operational control of bi-directional AGV system*, (1993), Int. J. Production Research, Vol. 31, pp. 2123-2138.

[30] Krishnamurthy N.N., Batta, R., Karwan, H., *Developing conflictfree routes for automated guided vehicles*, (1993), Operation research, Vol. 41, pp. 1077-1090.

[31] Moorthy, R.L., Hock-Guan, W., Wing-Cheong, N., Chung-Piaw, T., *Cyclic deadlock prediction and avoidance for zone-controlled AGV system*, (2003). Int. J. Production Economics, Vol. 83, pp. 309-324.

[32] Fanti, M. P., *Event-based controller to avoid deadlock and collisions in zone-control AGVS*, (2004), Int. J. Production Research, Vol. 40, pp. 1453-1478.

[33] Reveliotis, S.A., *Conflict resolution in AGV systems*, (2000), IIE Transactions, Vol. 32, pp. 647-659.

[34] Maza, S., Castagna, P., *Conflict-free AGV routing in bi-directional networks*, (2001), IEEE Int. conf. Emerging Technologies and Factory Automation, France, pp. 761-764.

[35] Maza, S., Castagna, P., Robust *conflict-free routing of bidirectional,* (2002), *AGVs*, IEEE Int. Conf. Systems, Man and Cybernetics,Tunisia.

[36] B. Lakehall, Y. Amirat, and J. Pontnau, *Fuzzy Steering Control of a Mobile Robot*, (1995), Industrial Automation and Control, Emerging Technologies, International IEEE/IAS, , pp. 383–386.

[37] S. Senoo, M. Mino, and S. Funabiki, *Steering Control of Automated Guided Vehicle for Steering Energy Saving by Fuzzy Reasoning*, (1992) Industrial Applications Society Annual Meeting, Conference Record of IEEE, pp. 1712–1716

[38] Wuwei C, Mills K, Wenwu S *A new navigation method for an automatic guided vehicle,* (2004) J Robot Syst 21(3):129–139. doi:10.1002/rob.20004

[39] Wu KH, Chen CH, Ko JM, *Path planning and prototype design of an AGV*, (1999*)*, J Math Comput Model 30:147–167. doi:10.1016/S0895-7177(99)00171-5

[40] Lin C-H, Wang L-L, *Intelligent collision avoidance by fuzzy logic control. Robot*, (1997), Auton Syst 20:61–83. doi:10.1016/ S0921-8890(96)00051-6

[41] Alves and Junior, *Mobile ultra sonic sensing in mobile robot*, (2002),In: IECON 02 28th Annual Conference of Industrial Electronics Society

[42] Sugeno M, Nishida M *Fuzzy control of model car. Fuzzy Sets*, (1985), Syst 16:103–113. doi:10.1016/S0165-0114(85)80011-7

[43] Sugeno et al, *Fuzzy algorithmic control of a model car by oral instruction. Fuzzy Sets*, (1989), Syst 32:207–219.doi:10.1016/0165-0114(89)90255-8

[44] M, Yahyaei., J. E. Jam , R. Hosnavi, Controlling the navigation of automatic guided vehicle(AGV) *using integrated fuzzy logic controller with programmable logic controller*, (2009), Springer-Verlag London, Vol4, PP. 795–807

[45] Prakash, A., Chen, M., *A simulaton study of flexible manufacturing systems*, (1995), Computers indo Engng, pp. 191-199.

[46] Xingl, B., Gaol, W., Battlel, K., Marwalal, T. *Can Ant Algorithms Make Automated Guided Vehicle System More Intelligent?* Faculty of Engineering and the Built Environment Universi ty of Johannesburg Johannesburg, South Africa.

[47] Lee, c., *Design and control of automated guided vehicle systems*, (2000), Department of Industrial and Manfuacturing Engineering, the Pennsylvania State Unviersity.

APPENDICES

Appendix A

```
 $regfile "m32def.dat"
 $crystal = 8000000
 Config Portb = Output
Config Lcdpin = Pin , Db4 = Portb.3 , Db5 = Portb.2 , Db6 = Portb.0 , Db7 = Portb.1 , E = Portb.4 , Rs = Portb.5
Config Lcd = 16 * 2
Config Adc = Single , Prescaler = Auto ,
Dim W1 As Word
Dim W2 As Word
Dim W3 As Word
Dim W4 As Word
Dim W5 As Word
Dim Channel1 As Byte
Dim Channel2 As Byte
Dim Channel3 As Byte
Dim Channel4 As Byte
Dim Channel5 As Byte
Channel1 = 0
Channel2 = 1
Channel3 = 2
Channel4 = 3
Channel5 = 4

Config Com1 = Dummy , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
'Deflcdchar 0 , 32 , 32 , 32 , 32 , 32 , 32 , 32 , 31      ' replace ? with number (0-7)$regfile "m32def.dat"
'Deflcdchar 1 , 32 , 32 , 32 , 32 , 32 , 32 , 31 , 31      ' replace ? with number (0-7)
'Deflcdchar 2 , 32 , 32 , 32 , 32 , 32 , 31 , 31 , 31      ' replace ? with number (0-7) $baud = 9600
'Deflcdchar 3 , 32 , 32 , 32 , 32 , 31 , 31 , 31 , 31      ' replace ? with number (0-7)
'Deflcdchar 4 , 32 , 32 , 32 , 31 , 31 , 31 , 31 , 31      ' replace ? with number (0-7)
'Deflcdchar 5 , 32 , 32 , 31 , 31 , 31 , 31 , 31 , 31      ' replace ? with number (0-7)
'Deflcdchar 6 , 32 , 31 , 31 , 31 , 31 , 31 , 31 , 31      ' replace ? with number (0-7)
'Deflcdchar 7 , 31 , 31 , 31 , 31 , 31 , 31 , 31 , 31      ' replace ? with number (0-7)

 Enable Interrupts
Start Adc
Dim A As Byte
Dim B As Byte
Config Portc = Output

Dim D0 As Long
Dim D As Long
Dim D1 As Long

Dim V As Long
Dim Vl As Long
Dim Vr As Long

Dim Mr As Long
Dim Ml As Long

Dim P As Long
Dim Temp As Long
Dim Delta As Long
Dim Mlf As Word
Dim Mlb As Word
Dim Mrf As Word
Dim Mrb As Word

Dim Direction As Integer
Dim Detectedjunction As Integer
```

```
Main:
  V = 70
  D0 = 200
  D1 = 600
  Mrf = 137
  Mrb = 139
  Mlf = 141
  Mlb = 143

  Direction = 1
  Detectedjunction = 0

Mystart:

  Do

  Gosub Readsensors
 '-------------------------------------------------

 If W1 < D1 And Direction > 0 Then
   Detectedjunction = Detectedjunction + 1
   Gosub Printscreen
   If Direction = Detectedjunction Then
      Gosub Turnright
      Gosub Mystart
   End If
 End If

 If W5 < D1 And Direction < 0 Then
   Detectedjunction = Detectedjunction - 1
   Gosub Printscreen
   If Direction = Detectedjunction Then
      Gosub Turnleft
      Gosub Mystart
   End If
 End If

 D = W4 - W2
 D = Abs(d)

 If D < D0 Then D = 0

 P = 0

 If D >= 800 Then P = 70
 If D >= 700 And D < 800 Then P = 60
 If D >= 600 And D < 700 Then P = 50
 If D >= 500 And D < 600 Then P = 40
 If D >= 400 And D < 500 Then P = 30
 If D >= 300 And D < 400 Then P = 20
 If D >= 200 And D < 300 Then P = 10

 Temp = W4 - W2

 If Temp < 0 Then P = P * -1
  Vl = V + P
  Vr = V - P
  Ml = Mlf
  Mr = Mrf
```

```
 If Vr > 127 Then
  Delta = Vr - 127
  Vr = 127
  Vl = -100                                  'Vl - Delta
 End If

 If Vl > 127 Then
  Delta = Vl - 127
  Vl = 127
  Vr = -100                                  'Vr - Delta
 End If

 If Vl < 0 Then
  Vl = Abs(vl)
  Ml = Mlb
 End If

 If Vr < 0 Then
  Vr = Abs(vr)
  Mr = Mrb
 End If

 Gosub Move

Loop

End                                        'end program

'---------------------------------------------------------------
'(
   Motors:

           Left Right
    Forward 141 137
    Backward 143 139

   Sensors:

    W1 R Right
    W2 Right
    W3 Center
    W4 Left
    W5 L Left
')
  Readsensors:
 W1 = Getadc(channel1)
 W2 = Getadc(channel2)
 W3 = Getadc(channel3)
 W4 = Getadc(channel4)
 W5 = Getadc(channel5)
 Gosub Printscreen
Return

Printscreen:
  Cls
  Locate 1 , 1
  Lcd W4

  Locate 1 , 6
  Lcd W3

  Locate 1 , 11
```

```
  Lcd W2

  Locate 2 , 1
  Lcd Detectedjunction

  Locate 2 , 3
  Lcd Direction

  Locate 2 , 5
  Lcd D

  Locate 2 , 9
  Lcd W1

  Locate 2 , 13
  Lcd W5
Return

 Move:
  Print Chr(mr) ; Chr(vr)
  Print Chr(ml) ; Chr(vl)
 Return                              'forwrd

 Turnright:
  Mr = Mrf
  Ml = Mlf
  Vr = V
  Vl = V

 Gosub Move

 Wait 1

  Mr = Mrb
  Ml = Mlf
  Vr = 80
  Vl = 127
 Wait 1
  Gosub Readsensors

  While W3 > D1
    Gosub Move
    Gosub Readsensors
  Wend

  Detectedjunction = 0

  Mr = Mrf
  Ml = Mlf
  Vr = V
  Vl = V
  Gosub Move
 Return                              'Turnright

 Turnleft:
  Mr = Mrf
  Ml = Mlf
  Vr = V
  Vl = V

 Gosub Move
```

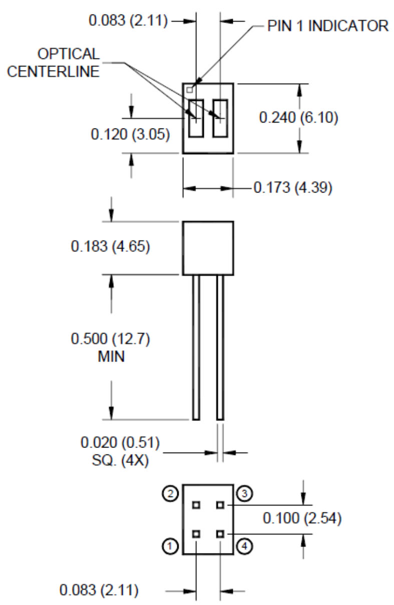
```
Wait 1

  Mr = Mrf
  Ml = Mlb
  Vr = 100
  Vl = 90
Wait 1
  Gosub Readsensors

  While W3 > D1
    Gosub Move
    Gosub Readsensors
  Wend

  Detectedjunction = 0

  Mr = Mrf
  Ml = Mlf
  Vr = V
  Vl = V
  Gosub Move
Return                                    'Turnleft

  Lifterup:
Reset Portc.1
Reset Portc.2

Set Portc.1
Reset Portc.2

'Wait 5

Return


  Lifterdown:
Reset Portc.1
Reset Portc.2

Set Portc.2
Reset Portc.1

'Wait 5

Return
```

Appendix B

FAIRCHILD SEMICONDUCTOR™

QRD1113/1114
REFLECTIVE OBJECT SENSOR

PACKAGE DIMENSIONS



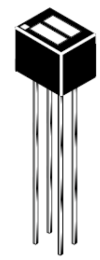
PIN 1 COLLECTOR PIN 3 ANODE
PIN 2 EMITTER PIN 4 CATHODE

NOTES:

1. Dimensions for all drawings are in inches (millimeters).
2. Tolerance of ± .010 (.25) on all non-nominal dimensions unless otherwise specified.
3. Pins 2 and 4 typically .050" shorter than pins 1 and 3.
4. Dimensions controlled at housing surface.

FEATURES

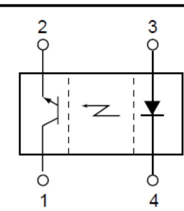
- Phototransistor Output
- No contact surface sensing
- Unfocused for sensing diffused surfaces
- Compact Package
- Daylight filter on sensor



NOTES (Applies to Max Ratings and Characteristics Tables.)

1. Derate power dissipation linearly 1.33 mW/°C above 25°C.
2. RMA flux is recommended.
3. Methanol or isopropyl alcohols are recommended as cleaning agents.
4. Soldering iron 1/16" (1.6mm) from housing.
5. As long as leads are not under any spring tension.
6. D is the distance from the sensor face to the reflective surface.
7. Cross talk (I_{CX}) is the collector current measured with the indicator current on the input diode and with no reflective surface.
8. Measured using an Eastman Kodak neutral white test card with 90% diffused reflecting as a reflective surface.

SCHEMATIC



ABSOLUTE MAXIMUM RATINGS (T_A = 25°C unless otherwise specified)

Parameter	Symbol	Rating	Units
Operating Temperature	T_{OPR}	-40 to +85	°C
Storage Temperature	T_{STG}	-40 to +85	°C
Lead Temperature (Solder Iron)(2,3)	T_{SOL-I}	240 for 5 sec	°C
Lead Temperature (Solder Flow)(2,3)	T_{SOL-F}	260 for 10 sec	°C
EMITTER			
Continuous Forward Current	I_F	50	mA
Reverse Voltage	V_R	5	V
Power Dissipation(1)	P_D	100	mW
SENSOR			
Collector-Emitter Voltage	V_{CEO}	30	V
Emitter-Collector Voltage	V_{ECO}		V
Power Dissipation(1)	P_D	100	mW

ELECTRICAL / OPTICAL CHARACTERISTICS (T_A = 25°C)

PARAMETER	TEST CONDITIONS	SYMBOL	MIN	TYP	MAX	UNITS
EMITTER Forward Voltage	I_F = 20 mA	V_F	—	—	1.7	V
Reverse Current	V_R = 5 V	I_R	—	—	100	µA
Peak Emission Wavelength	I_F = 20 mA	λ_{PE}	—	940	—	nm
SENSOR Collector-Emitter Breakdown	I_C = 1 mA	BV_{CEO}	30	—	—	V
Emitter-Collector Breakdown	I_E = 0.1 mA	BV_{ECO}	5	—	—	V
Dark Current	V_{CE} = 10 V, I_F = 0 mA	I_D	—	—	100	nA
COUPLED QRD1113 Collector Current	I_F = 20 mA, V_{CE} = 5 V D = .050" (6,8)	$I_{C(ON)}$	0.300	—	—	mA
QRD1114 Collector Current	I_F = 20 mA, V_{CE} = 5 V D = .050" (6,8)	$I_{C(ON)}$	1	—	—	mA
Collector Emitter Saturation Voltage	I_F = 40 mA, I_C = 100 µA D = .050" (6,8)	$V_{CE\ (SAT)}$	—	—	0.4	V
Cross Talk	I_F = 20 mA, V_{CE} = 5 V, E_E = 0 (7)	I_{CX}	—	.200	10	µA
Rise Time	V_{CE} = 5 V, R_L = 100 Ω	t_r	—	10	—	µs
Fall Time	$I_{C(ON)}$ = 5 mA	t_f	—	50	—	µs

TYPICAL PERFORMANCE CURVES

Fig. 1 Forward Voltage vs. Forward Current

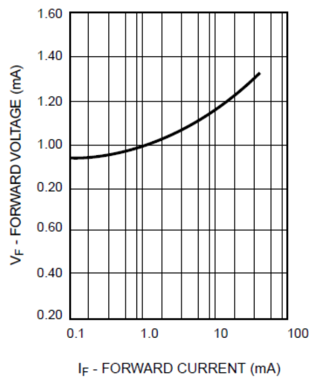


Fig. 2 Normalized Collector Current vs. Forward Current

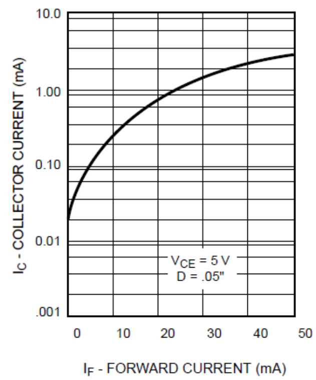


Fig. 3 Normalized Collector Current vs. Temperature

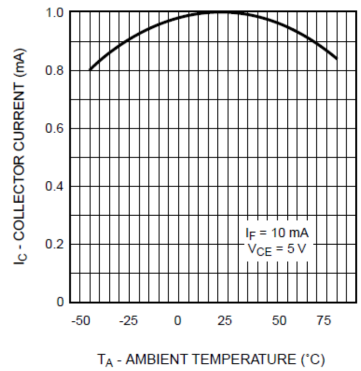


Fig. 4 Normalized Collector Dark Current vs. Temperature

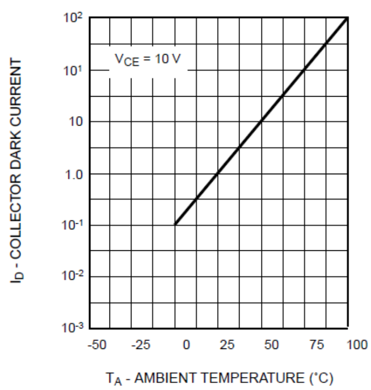
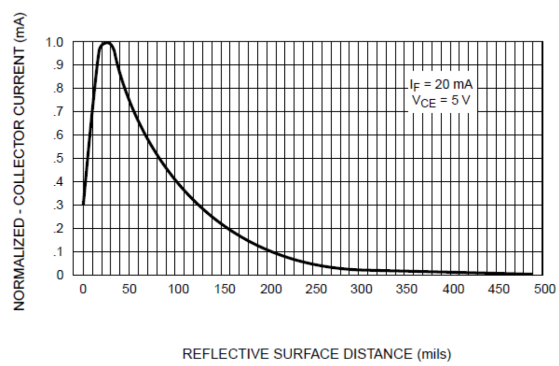


Fig. 5 Normalized Collector Current vs. Distance



DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Appendix C

Qik 2s12v10 User's Guide

1. Overview

The qik 2s12v10 adds a comprehensive yet easy-to-use, high-power option to Pololu's line of motor controllers. The compact board—it's almost the same size as the **dual VNH2SP30 carrier board** [http://www.pololu.com/catalog/product/708] by itself—allows any microcontroller or computer with a serial port to drive two brushed DC motors with full direction and speed control, providing up to 13 A (continuous) per motor channel and tolerating peaks as high as 30 A. The improvements over the previous generation and competing products include:

- high-frequency PWM to eliminate switching-induced motor shaft hum or whine
- a robust, high-speed communication protocol with user-configurable error condition response
- visible LEDs and a demo mode to help troubleshoot problematic installations
- reverse power protection on the power supply

Main Features of the Qik 2s12v10

- Simple bidirectional control of two DC brush motors.
- 6 V to 16 V motor supply range.
- 13 A maximum continuous current per motor (30 A peak).
- Logic-level, non-inverted, two-way serial control for easy connection to microcontrollers or robot controllers.
- RS-232-level, one-way serial control for easy connection to a PC serial port.
- Optional automatic baud rate detection from 1200 bps to 115.2 kbps.
- Seven on-board indicator LEDs (power, status/heartbeat, error indicator, and motor indicators) for debugging and feedback.
- Error output to make it easier for the main controller to recover from an error condition.
- Jumper-enabled demo mode allowing initial testing without any programming.
- Optional CRC error detection eliminates serial errors caused by noise or software faults.
- Optional motor shutdown on error or serial timeout for additional safety.

Specifications

Motor channels:	2
Motor supply voltage:	6 – 16 V
Continuous output current per channel:	13 A
Peak output current per channel:	30 A
Auto-detect baud rate range:	1200 – 115,200 bps
Available fixed baud rates:	115,200 bps, 38,400 bps, 9600 bps
Available PWM frequencies:	19.7 kHz, 9.8 kHz, 2.5 kHz, 1.2 kHz, 310 Hz, 150 Hz
Reverse voltage protection?:	Y
Motor driver:	VNH2SP30 x2

Important safety warning

This product is not intended for young children! Younger users should use this product only under adult supervision. By using this product, you agree not to hold Pololu liable for any injury or damage related to the use or to the performance of this product. This product is not designed for, and should not be used in, applications where the malfunction of the product could cause injury or damage. Please take note of these additional precautions:

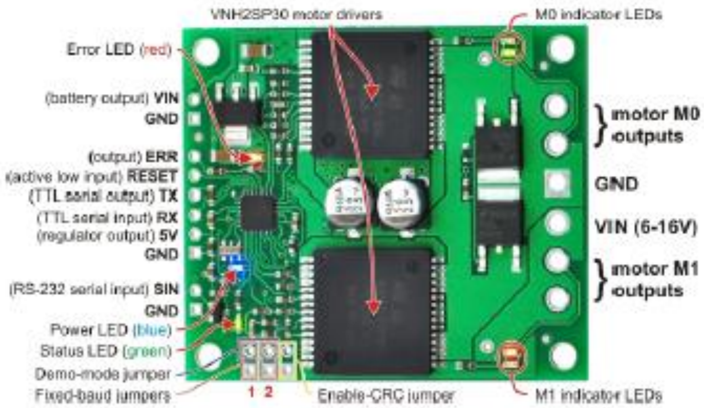
- This product contains lead, so follow appropriate handling procedures, such as not licking the product and washing hands after handling.
- Since the PCB and its components are exposed, take standard precautions to protect this product from ESD (electrostatic discharge), which could damage the on-board electronics. When handing this product to another person, first touch their hand with your hand to equalize any charge imbalance between you so that you don't discharge through the electronics as the exchange is made.
- Review the instructions carefully before making any electrical connections, and do all wiring while the power is turned off. Incorrect or reversed wiring could cause an electrical short or unpredictable behavior that damages this product and the devices it is connected to.
- This product is designed to be connected to motors, which should be operated safely. Wear safety glasses, gloves, or other protective equipment as appropriate, and avoid dangerous situations such as motors spinning out of control by designing appropriate safeguards and limits into your projects.

2. Contacting Pololu

You can check the **qik 2s12v10 dual serial motor controller page** [http://www.pololu.com/catalog/product/1112] for additional information. We would be delighted to hear from you about any of your projects and about your experience with the qik motor controller. You can **contact us** [http://www.pololu.com/contact] directly or post on our **forum** [http://forum.pololu.com/]. Tell us what we did well, what we could improve, what you would like to see in the future, or anything else you would like to say!

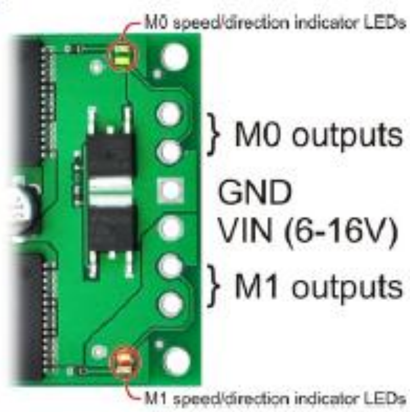
3. Connecting the Qik

Connecting to the qik can be as simple as hooking up power, your motors, and your serial connections. Many applications can ignore the jumpers and leave the remaining logic connections unconnected. The qik's serial transmit line, TX, is only necessary if you want to get feedback from the controller.



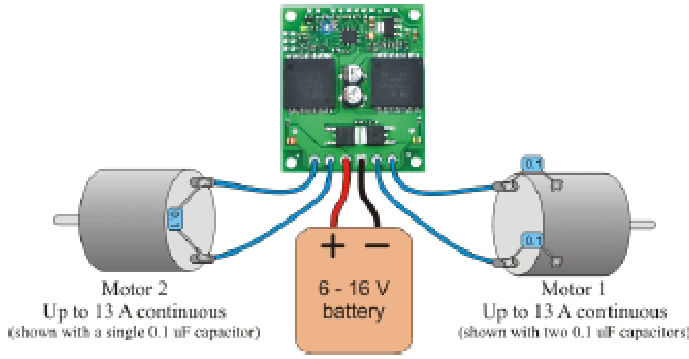
The qik logic and power connections and key components are shown above, and pins are labeled on the back side of the motor controller. All square pads are ground.

3.a. Power and Motor Connections



Power

The qik motor controller is powered via the large **VIN** and **GND** pads on the power side of the board, as shown in the picture above. The input voltage can be between 6 and 16 V and is the voltage that the motors will see. An integrated voltage regulator produces the 5 V that powers the board's logic, so no separate logic power supply is necessary. Both the input voltage and regulated voltage can be accessed as outputs on the left side of the board (i.e. the logic connections). See **Section 3.b** for more information. Please ensure that your power source can supply the current your motors will draw.



Qik 2s12v10 power and motor connections.

Motors

The qik can independently drive up to two bidirectional brushed DC motors, referred to as M0 and M1. The two terminals of each motor should be connected to the qik as shown above. Variable speed is achieved with 7-bit or 8-bit pulse width modulated (PWM) outputs at one of several selectable frequencies. 7-bit control allows for PWM frequencies of 19.7 kHz, 2.5 kHz, and 310 Hz; 8-bit control allows for PWM frequencies of 9.8 kHz, 1.2 kHz, and 150 Hz. The highest achievable frequency of 19.7 kHz is ultrasonic, which can result in quieter motor control. Lower frequencies might produce louder motor noise, but they can help decrease power losses due to switching. The resolution and frequency can be set via the qik's PWM configuration parameter (see **Section 5.a**).

The motor direction convention used in this document is that "forward" corresponds to holding the + output at VIN while PWMing the - output between ground and high impedance. "Reverse" is the same as forward but with the outputs flipped: - is held at VIN while + is PWMed between ground and high impedance. As a result, the motor is rapidly alternating between drive and coast when the direction is "forward" or "reverse". Variable speed control is achieved by varying the fraction of the cycle that the motor outputs are driving. Full speed arises when the motor outputs are driving 100% of the time (one motor output is held at VIN and the other at ground). See **Section 5.f** for more information.

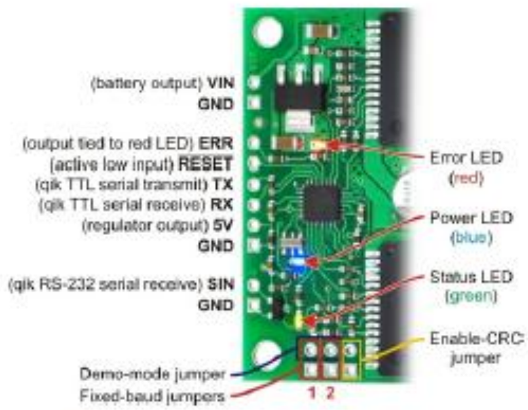
The qik 2s12v10 allows for variable braking. In this mode, the motor's + and - outputs are PWMed between ground and high impedance. While the outputs are high-impedance, the motor coasts, and while the outputs are tied to ground the motor brakes. See **Section 5.e** for more information.

The qik 2s12v10 motor controller uses VNH2SP30 motor driver integrated circuits. These motor drivers have maximum current ratings of 30 A continuous, but the chips by themselves will overheat at lower currents (see the table below for typical values). The actual current the qik can deliver will depend on how well the motor drivers are kept cool. The qik's printed circuit board is designed to draw heat out of the motor driver chips, but performance can be improved by adding a heat sink. In our single-driver tests, we were able to deliver 30 A for a fraction of a second and 20 A for several seconds without overheating the IC. At 6 A, the chip gets just barely noticeably warm to the touch. For high-current installations, the motor and power supply wires should also be soldered directly instead of going through the supplied terminal blocks, which are rated for up to 15 A.

- Time to overheat at 30 A: < 1 s
- Time to overheat at 20 A: 35 s
- Time to overheat at 15 A: 150 s
- Time to overheat at ≤ 13 A: N/A (does not overheat)

Note that these above times were obtained using only one driver with 100% duty cycle at room temperature without a heat sink. Drawing high currents from both drivers simultaneously could cause them overheat faster. Switching-induced power losses arising from duty cycles below 100% could also cause the drivers to overheat faster or lower the continuous current rating.

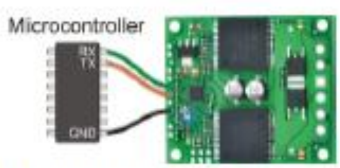
3.b. Logic Connections

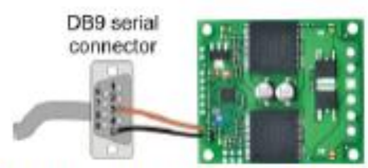


Serial Lines: RX, TX, and SIN

The qik can accept a logic-level (0 – 5 V), non-inverted serial input connected to its serial receive line, **RX**, and it can handle baud rates from 1200 – 115,200 bps. This type of serial is often referred to as TTL and is an interface method commonly used by microcontrollers. The voltage on this pin should not exceed 5 V. The qik provides logic-level, non-inverted serial output on its serial transmit line, **TX**, in response to commands that request information. Information requests always result in the transmission of a single byte per request. If you aren't interested in receiving feedback from the qik, you can leave this line disconnected.

The qik can also accept RS-232 serial input connected to the serial receive line, **SIN**. A computer serial port typically communicates via RS-232 serial, which is inverted and uses voltages that would be out of spec for the rest of the qik's inputs (e.g. -12V to 12V), so SIN is the only pin to which it is safe to make a direct RS-232 connection. The qik does not have an RS-232 output, so you will need to use an RS-232 level converter connected to the logic-level output if you need RS-232 feedback from the qik.





Qik 2s12v10 TTL serial connection example (transmit and receive).Qik 2s12v10 RS-232 serial connection example (qik receive only).

Both RX and SIN connect to the same serial port on the qik, so you should not use both of these inputs simultaneously. Also, don't forget to connect your serial source's ground to the qik's ground!

Reset

The reset line, **RST**, is an active-low input, which means that it will reset the qik when driven low. This pin is internally pulled high, so many applications can leave this pin disconnected.

Error

The error line **ERR** is an output that is connected to the red error LED and will drive high in response to an error (which in turn lights the LED). Once an error occurs, the pin outputs high until a serial command is issued to read the error byte, at which point the pin goes to a high-impedance state that is pulled low through the LED. This allows you to connect the error lines of multiple qiks to the same digital input. For more information on the possible error conditions, please see **Section 5.c**. If you don't care about error detection, you can leave this pin disconnected.

5V (out)

This line connects to the 5 V output of the qik's voltage regulator and can be used to power additional electronics in your system. It can safely supply up to 70 mA beyond what the board draws when VIN is 16 V. The closer the input voltage is to 5 V, the more current the regulator can deliver without overheating.

VIN (out)

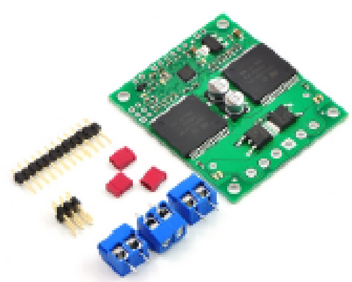
This is a convenient connection point to the input voltage that can be used as a power source for additional electronics. Note that this pin is not intended to handle high currents, so **it should not be used to power the qik** (use the large VIN and GND pads on the other side of the board for this). Do not attempt to draw more than 1 A from this pin.

Connecting to a 3.3 V Microcontroller

The logic components on the qik 2s12v10 run at 5 V, but it is still possible to interface with a 3.3 V microcontroller. The **RX** high input threshold is 3 V, so you can directly connect your microcontroller's transmit line to the qik's TTL serial receive line (i.e. no additional components are required for sending commands to the qik from a 3.3 V MCU).

If your microcontroller digital inputs are 5V-tolerant, you can make direct connections to the **ERR** and **TX** outputs and **RST** input, which is weakly pulled to 5 V on the qik. If not, you can leave these optional outputs unconnected, or you can use external components to decrease the voltage to a range your MCU can handle. A simple way to accomplish this is by placing voltage dividers between each qik output and your MCU.

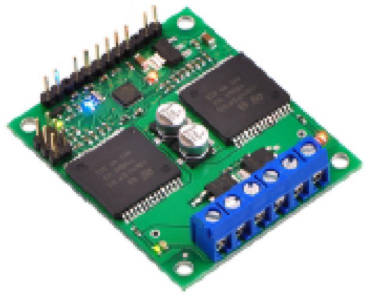
3.c. Included Hardware



The qik ships with a 12×1 **straight 0.100" male header strip** [http://www.pololu.com/catalog/product/965], a 3×2 **straight 0.100" male header strip** [http://www.pololu.com/catalog/product/966], three **2-pin terminal blocks** [http://www.pololu.com/catalog/product/830], and three **red shorting blocks** [http://www.pololu.com/catalog/product/971].

For the most compact installation, you can solder wires directly to the qik pins themselves and skip using the included hardware. For high-current installations, you should avoid using the supplied terminal blocks, which are rated for up to 15 A, and instead directly solder the motor and power supply wires to the pads.

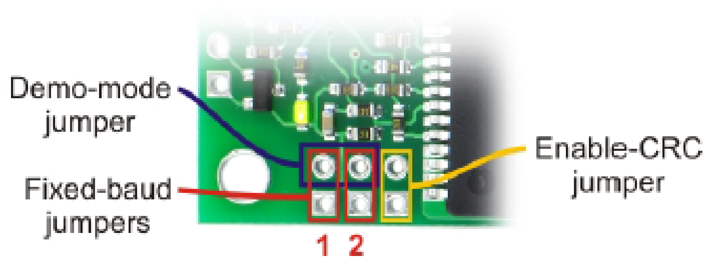
The included hardware allows you to make less permanent connections. You can break the 12×1 header strip into a 6×1 piece and two 2×1 pieces and solder these strips into the qik's logic pins where you plan on making connections, or you can use a pair of pliers to pull out the two header pins in the original 12×1 strip for which the qik has no holes and solder the entire strip to the qik's logic pins. You can see this latter approach in the picture below. You can then make your own cables that have **female headers** [http://www.pololu.com/catalog/category/50] on them and plug these onto the male headers on your qik, or you can solder the pins to the other side of the board and simply plug your qik into a breadboard. You might also consider using a **0.100" right-angle male header strip** [http://www.pololu.com/catalog/product/967] (not included) for a lower profile.



The 3×2 header strip can be soldered to the jumper pins as shown above, which lets you make use of the included shorting blocks, and the included terminal blocks lock together to make a single, 6-pin strip that you can solder to the power side of the board.

3.d. Jumpers

The qik jumpers allow you to easily alter the behavior of the device. These jumpers can be left off for most applications. If you use a jumper, it must be in place when the unit first starts up; changing the jumpers while the unit is running does not take effect until the qik is reset or power is cycled. The only exception to this is the removal of the demo mode jumper while the qik is in demo mode, which takes the qik out of demo mode.



Fixed-Baud Modes

The jumpers labeled **BAUD1** and **BAUD2** on the bottom of the qik (i.e. the two closest to the logic connection side of the board) can be used to set the qik to fixed-baud mode when a shorting block is in place across one or both jumper locations. When neither of these jumper locations has a shorting block, the qik is in auto-detect mode and determines the baud rate automatically when it receives the first 0xAA (170 in decimal) byte. If you have a noisy serial connection or find that the automatic baud detection is not working well for your application, you can use a shorting block or some other jumper to ground pins BAUD1 and/or BAUD2 (the circular pads right next to the silkscreen labels). This sets the baud rate to a predetermined value, as described in the table below, and the qik skips the automatic baud detection phase that normally occurs on start-up.

BAUD1 Jumper	BAUD2 Jumper	Baud Mode
OFF	OFF	auto-detect baud rate (1200 – 115,200 bps)
ON	OFF	fixed baud rate at 115,200 bps
OFF	ON	fixed baud rate at 38,400 bps
ON	ON	fixed baud rate at 9,600 bps

Enable-CRC Mode

The jumper labeled CRC on the bottom of the qik enables cyclic redundancy check (CRC) mode when the shorting block is in place. This allows you to increase the robustness of your qik connection through the addition of a CRC error-checking byte to the end of the command packets you send to the qik. The default behavior of the qik is to simply respond to a command packet once it receives the last byte. Grounding pin CRC (the circular pad right next to the "CRC" silkscreen label) causes the qik to expect an additional byte at the end of the command byte that results from a CRC-7 computation on the entire message. If this byte does not match the expected CRC, the qik ignores the command and uses the ERR pin to announce a CRC error. Please see **Section 6** for more information on how CRC error detection works. When in this mode, the green heartbeat LED will flicker twice per heartbeat rather than the single flash that occurs when CRC error checking is disabled.

Demo Mode

If you short pin **BAUD1** to pin **BAUD2** (the circular pads right next to the silkscreen labels) and reset the qik, it enters demo mode and remains in demo mode for as long as the short is maintained. Demo mode gives you an easy way to test your qik and troubleshoot your application for potential problems. In demo mode, the qik smoothly ramps motor M0 from stopped to full-speed forward to full-speed reverse to stopped again over a few seconds. It then does the same for motor M1. While motor M0 is active, the red LED is on; while motor M1 is active, the red LED is off. While a motor is being driven forward, the green LED is on; while a motor is being driven in reverse, the green LED is off.

Demo mode can help you determine before you've even written any code if you have an issue with your power supply, such as insufficient ability to supply the current your motors are drawing or interference from motor noise.

While in demo mode, any serial data that is received by the qik on the RX line is echoed on the TX line, giving you an easy way to test your serial connection. It accomplishes this by repeatedly checking the state of the serial input pin and setting the state of the output pin to match. This process works reliably at low baud rates, but it is not fast enough to keep up well at higher baud rates, which can result in an echoed byte that does not match the one you transmitted.

3.e. Indicator LEDs and Phases of Operation

LED Overview

The qik has seven LEDs that are used to provide feedback about its state of operation:

- **Power LED** (blue): This blue LED is lit when the board is receiving power. It is located on the logic side of the board.
- **Status LED** (green): This green LED provides a heartbeat to let you know that your qik is alive and what state the qik is in, and it also acts as a serial activity indicator. It is is located on the logic side of the board near the jumpers.
- **Error LED** (red): This red LED is tied to the **ERR** output pin and lights in response to an error (which drives the ERR pin high). Once an error occurs, the LED remains lit until a serial command is issued to read the error byte, at which point the LED turns off. For more information on the possible error conditions, please see **Section 5.c**. It is located on the logic side of the board.
- **Motor Indicator LEDs**: Each motor has a green and red indicator LED that are tied directly to the motor driver outputs, for a total of four motor indicator LEDs. The color of the lit LED tells you the direction the

motor driver would turn an attached motor (green corresponds to "forward"), and the brightness of the LED gives you feedback about the speed. Please note that when a motor is attached and driven at an intermediate speed, it is normal to see both the red and green LED lit simultaneously. This is because the inductance of the motor keeps the current flowing during the coast phase of the PWM cycle, and since it cannot flow back through the driver, the only path it has is back through the LED that corresponds to the opposite direction. These LEDs are located on the power side of the board.

Automatic Baud Detection Phase

When the qik first starts up in automatic baud detection mode, it enters a phase in which it is waiting to receive the byte 0xAA at a baud rate that is within the range of 1200 bps to 115,200 bps. If the serial receive line **RX** is pulled high, as is expected for an idle TTL serial line and the default state of an unconnected RX, the green status LED fades in and out evenly every 0.8 s. If the serial receive line RX is held low, this is indicative of a bad serial connection and the red error LED cycles between being on for 0.4 s and off for 0.4 s (and the green status LED is off).

If a serial byte other than 0xAA is received in this mode, or if 0xAA is transmitted at an invalid baud rate, the red error LED turns on and stays on until the automatic baud detection phase ends. This gives you feedback that the baud has not yet successfully been set and you are still in the automatic detection phase. Once the baud is detected, this phase ends and the qik proceeds to normal operation.

Normal Operation

In normal operation, the green status LED very briefly flashes a heartbeat every 1.3 seconds. If the qik is in enable-CRC mode, the heartbeat LED flickers twice in rapid succession; if not, it pulses just once per heartbeat. If any serial activity is detected, the green status LED turns on until the next heartbeat turns it off. If an error occurs, the red error LED turns on and remains on until you issue a get-error command (see **Section 5.c**).

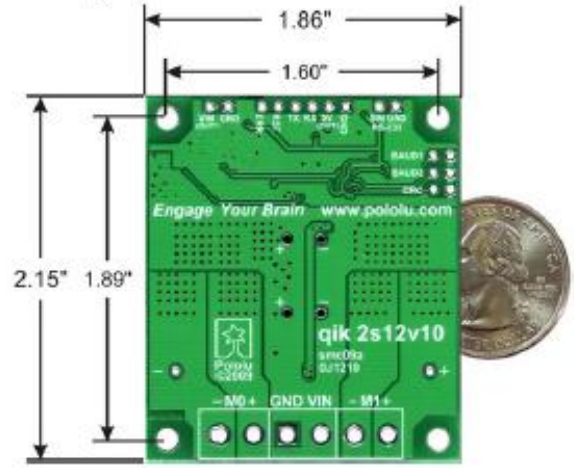
Demo Mode

In demo mode, the status and error LEDs cycle through the pattern:

1. red (error) and green (status) on
2. red (error) on and green (status) off
3. red (error) off and green (status) on
4. red (error) and green (status) off

This cycle takes five seconds, and each of the four LED states corresponds to a different output state of the qik's motor ports. When the red error LED is on, motor M0 is active, and when the red error LED is off, motor M1 is active. When the green status LED is on, the active motor is moving forward, and when the green status LED is off, the active motor is moving in reverse. If you don't have any motors connected, the motor indicator LEDs light according to this pattern, and you can see them fade in and out as the motor speed ramps up and back down.

3.f. Board Dimensions and Mounting Information



The qik 2s12v10 measures 2.15" x 1.86" x 0.28" (54.6 x 47.2 x 7.1 mm) and weighs 0.5 oz (14 g) without the header pins or terminal blocks installed. It has four mounting holes, one in each corner. The holes have a diameter of 0.125" and are designed for #4 or M3 screws.

Appendix D

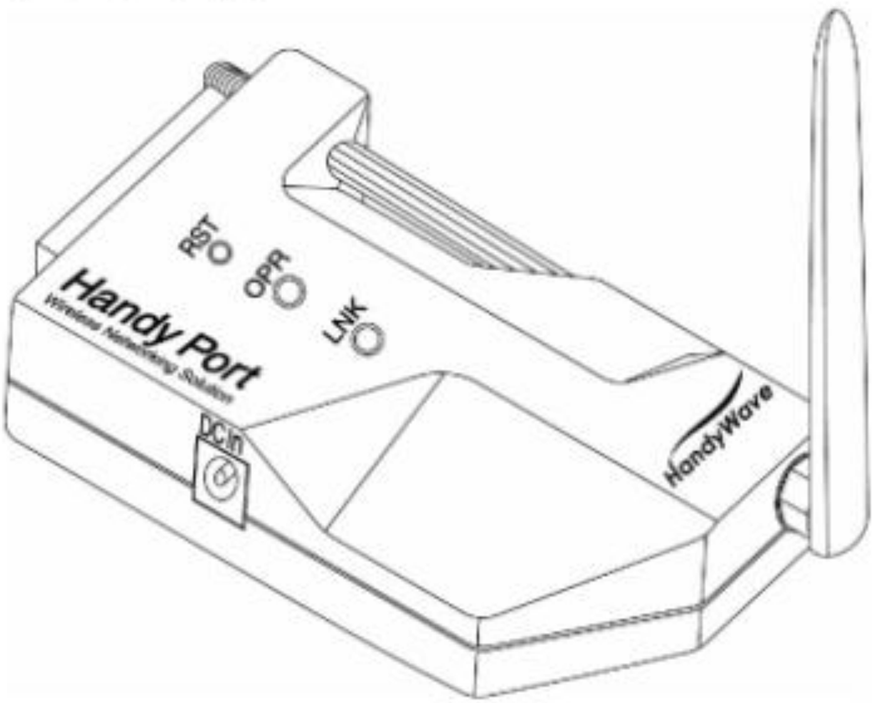
HPS-120

HandyPort-Serial

Wireless Solutions in your Hand

User's Manual

Version 2.0



HandyWave

1. Introduction

Thank you for purchasing a HandyPort-Serial. The HandyPort-Serial can be used as a component in many types of systems allowing them to communicate wirelessly with other Bluetooth products such as PC-cards, laptops, handheld computers, mobile phones and other HandyPort-Serial. The HandyPort-Serial is a suitable component in new products as well as in existing products.

1.1. Features

- Supports Bluetooth Serial Port Profile and Generic Access Profile
- No need of external host and software
- Easy of installation and use
- Supports configuration of the local device
- Supports configuration of the remote device via Over-the-Air
- Easy of maintenance
- Supports up to 100 meter (Line of Sight)

1.2. Package

- HPS-120 2 EA
- Antenna 2 EA
- A USB Cable for Power Supply
- A Manual[1]

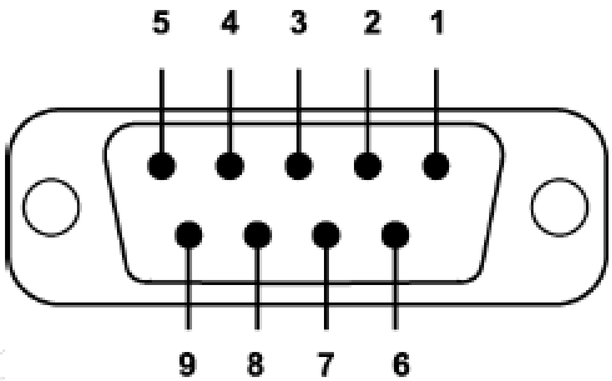
2. Specifications

2.1. General

Baud Rate	Up to 115.2kbps (Recommend above 2.4kbps) Supports 1.2/2.4/4.8/9.6/19.2/38.4/57.6/115.2kbps
Coverage	Up to 100 M
Connection	Point-to-Point
Signal	DCD, TxD, RxD, GND, CTS/DSR[1], DTR, RTS
RS-232 Interface	D_SUB 9 Pin Female
Standard	Bluetooth Specification Version 1.1
Frequency	2.400 ~ 2.4835GHz
Hopping	1,600/Sec, 1MHz Channel Space
Modulation	GFSK, 1Mbps, 0.5BT Gaussian
Tx. Power	Max 20 / Typical 16dBm (Class 1)
Rx. Sensitivity	-84dBm
Antenna Interface	SMA Female
Antenna Gain	Max. 2dBi
Power Supply	+5 ~ +12Vdc
Current Consumption	Max. 110mA
Operation Temperature	-20 ~ 75 °C
Size	35mm (W) x 65mm (D) x 16mm (H)

2.2. RS-232 Interface

2.2.1. Pin-out



2.2.2. Signals

Pin Number	Signal	Direction	Description
1	DCD	Output	Data Carrier Detect
2	TxD	Output	Transmitted Data
3	RxD	Input	Received Data
4	DSR	N/A (Input)	Option: Data Set Ready[1]
5	GND	N/A	Signal Ground
6	DTR	Output	Data Terminal Ready
7	CTS	Input	Clear to Send
8	RTS	Output	Request to Send
9	Vcc	Input	Power Supply

2.3. Factory Settings

The following is the factory settings of COM port. You can change the factory settings of COM port with commands. In this case, you have to remember the changed factory settings.

- Baud rate: 9600 bps
- Data Bit: 8 bit
- Parity Bit: No parity
- Stop Bit: 1 stop bit
- Flow control: None

2.4. Status LED

There are two LED on HPS-120.

- OPR (Red): When HPS-120 is supplied the power, it is turned on or flashed.
- LNK (Green): When a wireless link is on, it is turned on. If HPS-120 is in the configuration mode, it will be flashing every second.

2.5. Reset Button

The RST button has the following functions.

- Enter / Exit the configuration mode
- Restore the factory settings[1]
- Disconnect and reconnect a wireless connection.

2.5.1. Entering the Configuration Mode

When the LNK LED is OFF, push the RST button. When the LNK LED is ON, you have to push the RST button twice to enter the configuration mode. If you enter the configuration mode successfully, LNK LED will be flashing every second. And HPS-120 COM port will be stored the factory settings.

2.5.2. Exiting the Configuration Mode

You can have two options to exit the configuration mode.

Exit the configuration mode by software: Type "X".

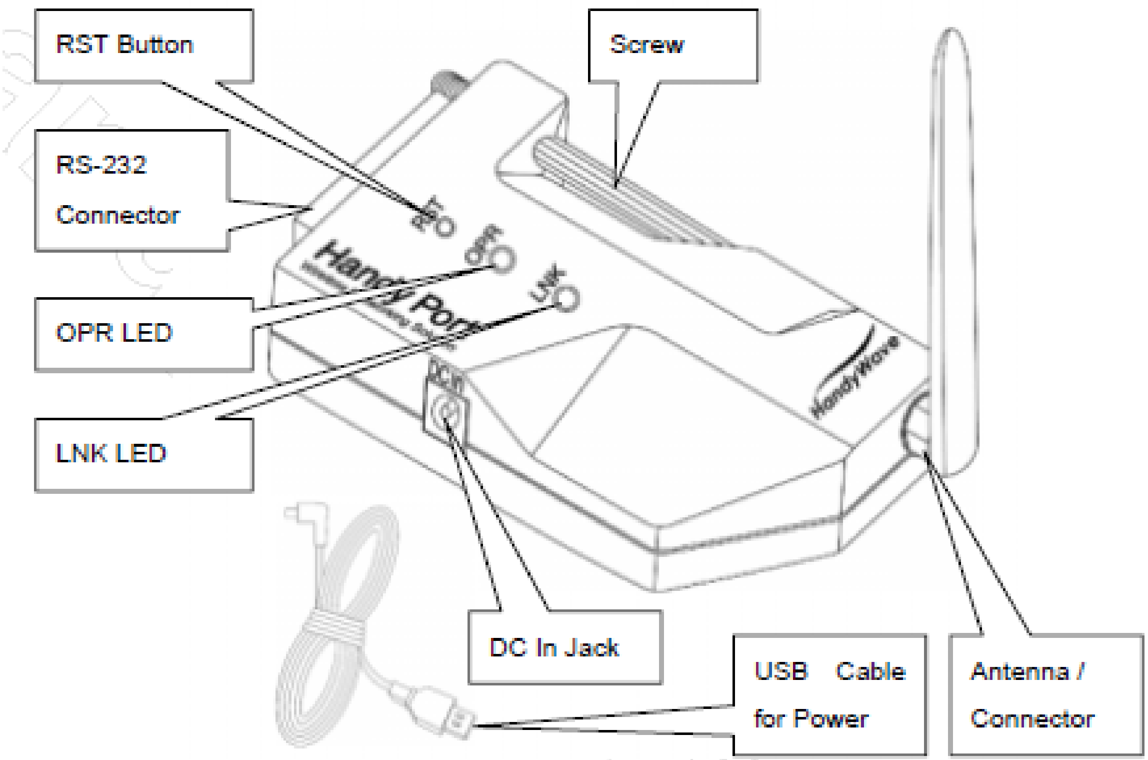
Exit the configuration mode by the RST button: Push the RST button.

2.5.3. Re-connection

When the LNK LED is on, you can push the RST button to disconnect and reconnect a wireless link.

3. Hardware Installation

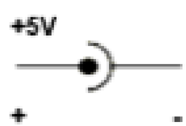
3.1. Hardware Description



3.2. Power Supply

You can supply power to the HPS-120 as follows:

- Use an AC/DC converter (Output Power: +5 ~ +12Vdc / 300mA).



- Use a provided USB cable.
- You can supply power via 9th pin of D_SUB 9 Pin connector.

3.3. Install Procedure

Step 1: Assemble a provided antenna to HPS-120 body.

Step 2: Plug a HPS-120 into the COM port of device.

Step 3: Power on.

Step 4: Configure the HPS-120, if necessary.

4. Usage

You can change the configuration of HPS-120 using Hyper Terminal[1].

4.1. Hyper Terminal Settings

Baud Rate: 9600 bps / Data Bit: 8 / Parity Bit: None / Stop Bit: 1 / Flow Control: None / Emulation: VT100

4.2. Configuration

4.2.1. Starting Configuration

Step 1: Plug a HPS-120 into a COM port of PC. And Power it on.

Step 2: Open a Hyper Terminal and set it up.

Step 3: Push the RST button on HPS-120. If you enter the configuration mode successfully, LNK LED will be flashing every second.

Step 4: Hit the <Enter> key, 5 second later.

Step 5: Change the configuration of HPS-120 with commands, if necessary.

4.2.2. Usage Printing

If you are in the configuration mode, type "?<Enter>" for listing of commands. If you want to know the usage of a command, type "?[command]<Enter>". All commands and parameters are case sensitive. And you cannot use a <Backspace>.

4.2.3. After Configuration

After finishing the configuration, you have to execute a command "X" to apply changes.

4.3. Command Set

The commands are as follows[1]:

Item	Syntax	Description	Remarks
1. Connecting address	AAddr<CR>	Set a remote device address for a wireless connection.	A local and remote BD_ADDR always need to be difference.
2. Baud rate	BBR[D]<CR>	Change the baud rate. D (option): Change a factory setting[2].	Baud Rate (BR) - 0: 1200, 1: 2400, 2: 4800, 3: 9600, 4: 19200, 5: 38400, 6: 57600, 7: 115200
3. COM port	CCOMPort<CR>	Change a request serial port.	COMPort: '1' ~ '7' Only valid in connection mode 2.
4. PIN code	EPIN<CR>	Authentication Off: hit <Enter> Authentication On: Type up to 11 characters	Paired adapters should have a same PIN code.
5. Flow control	FFC[D]<CR>	Set the Flow control. D (option): Change a factory setting[3].	FC - 0: None 1: Hardware[4] 2: DTR/DSR[5]
6. Search timer	GTO<CR>	Set a search timeout. TO (timeout): ASCII '0' ~ "999"	Connection mode 3 only. Default: 10 sec.
7. Max number of search	HNO<CR>	Set the max number of search. NO: ASCII '0' ~ "999"	Connection mode 3 only. Default: 10
8. Search device	ITO,NO[L]<CR>	Execute searching devices. TO: ASCII '0' ~ "999" NO: ASCII '0' ~ "999" L (option): Display a long form.	Connection mode 3 only. ',': ASCII 0x2C
9. Discovery mode	JE/D<CR>	Set the discovery mode. 'E': Enable 'D': Disable	Connection mode 1 only. Default: Enable
10. Low Power Mode	KE/D<CR>	Set the low power mode. 'E': Enable 'D': Disable	Default: Disable

11. Connection mode	MMode<CR>[1]	Set a connection mode. Mode: '0' – '3' Mode 0 & 2: Required a remote address. Mode 2: Required a serial port.	0: 1:1 Mode 1: WAIT Mode 2: REGISTER and CONNECT Mode 3: WAIT Command Mode
12. Friendly name	NName<CR>	Set a friendly name up to 11 characters.	
13. Parity Bit	PPA[D][2]<CR>	Set the parity bit. D (option): Change a factory setting[3].	0: None, 1: Odd 2: Even
14. Connection Timeout	QTO<CR>	Set the connection timeout. TO: ASCII '0' ~ "999"	Connection mode 3 only. Default: 10 sec.
15. Stop Bit	SST[D]<CR>	Set the stop bit. D (option): Change a factory setting[4].	0: 1 Stop, 1: 2 Stop
16. Connect	TAddr[,TO]<CR>	Try to make a connection. Addr: a remote address TO (option): ASCII '0' ~ "999"	Connection mode 3 only. ',': ASCII 0x2C Default Timeout: 10 sec.
17. Cancel	U	Cancel a command.	Connection mode 3 only.
18. View	V	Display the device information	You can find out a software version.
19. CoD	WCoD<CR>	Set the class of device. CoD: 6-Hex in ASCII	Default: "001F00"
20. Exit	X	Apply changes.	Rebooting
21. Status	Z	Display the status of state machine.	'S': Idle / 'P': Pairing / 'C': Connecting / 'A': RF on / 'I': Inquiring
22. Usage	?[C]<CR>	Display the command list or usage. C: Command	AT+Z?<CR>: Command list AT+Z?A<CR>: Usage of 'A'