DEVELOPMENT OF A MINIATURIZED AUTOMATED PRODUCTION CONTROL SYSTEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZGÜR KORKMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JANUARY 2012

Approval of the thesis:

DEVELOPMENT OF A MINIATURIZED AUTOMATED PRODUCTION CONTROL SYSTEM

submitted by **ÖZGÜR KORKMAZ** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Süha Oral ____________________
Head of Department, **Mechanical Engineering**

Prof. Dr. S. Engin Kılıç ____________________
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Can Çoğun ____________________
Mechanical Engineering Dept., METU

Prof. Dr. S. Engin Kılıç ____________________
Mechanical Engineering Dept., METU

Prof. Dr. Mustafa İlhan Gökler ____________________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. A. Buğra Koku ____________________
Mechanical Engineering Dept., METU

Prof. Dr. Sinan Kayalıgil ____________________
Industrial Engineering Dept., METU

Date: ____________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : **Özgür Korkmaz**

Signature :

ABSTRACT

DEVELOPMENT OF A MINIATURIZED AUTOMATED PRODUCTION CONTROL SYSTEM

Korkmaz, Özgür
M.Sc., Department of Mechanical Engineering
Supervisor : Prof. Dr. S. Engin Kılıç

January 2012, 90 pages

In this thesis a custom embedded system and control software developed for an Automated Storage and Retrieval System (AS/RS) which is based on the Computer Integrated Manufacturing Laboratory (CIMLAB) located in the Department of Mechanical Engineering, Middle East Technical University. Primary objective of this study is AS/RSs related control rules can be applicable to the current physical system. The secondary objective is determined as developing the control system in a flexible way that allows adding new equipments to the system and configuring parts of the system.

Two types of control board are manufactured and also boards' firmware and computer software are developed. These two boards communicate with computer one at a time. Some AS/RS related control rules are implemented at the control software. According to these rules the control software assigns tasks to the related board. Also the control software records necessary information in order to measure the performance of the AS/RS.

Several control rules as storage assignment, dwell point and sequencing of storage and retrieval order rules are applicable to the AS/RS without need for low level programming. Because of the physical limitation, batching rules cannot be applied to the current system. Also a graphical user interface is developed for using the system easily and observing the real time status of the system equipments.

Two experiments are designed and run in order to show flexibility of the control system. Different control rules applied to each of the experiment. Experiment results put forth the control system was quite successful in meeting the objectives.

Keywords: Automated Storage and Retrieval Systems, Control Software for AS/RS, Physical Simulation in Production Management

ÖZ

MİNYATURİZE EDİLMİŞ OTOMATİK ÜRETİM KONTROL
SİSTEMİNİN GELİŞTİRİLMESİ

Korkmaz, Özgür
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. S. Engin Kılıç

Ocak 2012, 90 sayfa

Bu tezde Orta Doğu Teknik Üniversitesi, Makina Mühendisliği Bölümünde bulunan Bilgisayar Tümleşik Üretim laboratuarındaki otomatik depolama sistemi için özel bir gömülü sistem ve kontrol yazılımı geliştirilmiştir. Bu çalışmanın birincil hedefi otomatik depolama sistemleri ile ilgili kontrol kurallarının mevcut fiziksel sisteme uygulanabilmesidir. İkincil hedef yeni ekipmanları sisteme eklemeye ve sistemin parçalarını yapılandırmaya izin veren bir kontrol sistemi geliştirmek olarak belirlenmiştir.

İki çeşit kontrol kartı üretilmiştir ve kartların aygıt yazılımı ile bilgisayar yazılımı geliştirilmiştir. Bu iki karttan her seferde bir tanesi bilgisayar ile haberleşir. AS/RS ile ilgili bazı kontrol kuralları kontrol yazılımında uygulanmıştır. Bu kurallara göre bilgisayar ilgili kartlara görevlere atamaktadır. Ayrıca kontrol yazılımı otomatik depolama sisteminin performansını ölçmek amacıyla gerekli bilgileri kaydeder.

Depolama konumu ataması, durma noktası ve depolama ve geri alma taleplerinin sıralanması gibi çeşitli kontrol kuralları düşük seviye programlamaya gerek olmadan otomatik depolama sistemine uygulanabilmektedir. Fiziksel kısıtlama nedeniyle gruplama kuralları mevcut sisteme uygulanamamaktadır. Ayrıca sistemin kolay kullanımı ve sistem ekipmanlarının gerçek zamanlı durumunu gözlemlemek için bir kullanıcı arayüzü geliştirilmiştir.

Kontrol sisteminin esnekliğini göstermek amacıyla iki deney tasarlanmış ve koşturulmuştur. Her bir deneyde farklı kontrol kuralllları uygulanmıştır. Deney sonuçları kontrol sisteminin hedefleri karşılamada oldukça başarılı olduğunu ortaya koymuştur.

Anahtar Kelimeler: Otomatik Depolama Sistemleri, Otomatik depolama sistemi için kontrol yazılımı, İmalat Yönetiminde Fiziksel Benzetim

to my family…

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to Prof. Dr. S. Engin Kılıç for his valuable efforts throughout my thesis. I am also grateful to Prof. Dr. Sinan Kayalıgil, Asst. Prof. Dr. A. Buğra Koku and Asst. Prof. Dr. Melik Dölen.

I would like to thank Yeşim Dildar for making the thesis period enjoyable and for her invaluable support.

Finally, I would like to express my sincere thanks to my family for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Today's competitive business environment gives importance to the efficient management of warehouse. Automated Storage and Retrieval System (AS/RS) is important part of the warehouse. After the introduction of AS/RS in the 1950s, they have been widely used in distribution and production environments. There are several benefits of AS/RS such as savings in labor costs, improved safety, improved throughput and utilized more floor space. The main disadvantage of AS/RS is high investment cost. Applications of AS/RS exist in clean room manufacturing environment in order to reduce contamination of product from manual handling, in healthcare environment, in cold environments as frozen food distribution centers, in manufacturing and distribution of electronic products and so forth. Generally, an AS/RS consists of racks, stackers, aisles and I/O points. Stackers, which run through aisles, serve racks. Pallets are used to store products to the racks and retrieve products from racks. At the I/O points, products enter the system or exit from it.

Effective management of AS/RS contributes to faster response times to customers and decreases overall costs in a warehouse. The efficient operation of AS/RS requires physical design and control design. Physical design of AS/RS consists of size of storage locations, length of aisles, location of I/O points, number of stackers and so forth. Control design is examined from four aspects as storage assignment, batching, dwell point and sequencing of storage and retrieval orders.

In METU-ME-Computer Integrated Manufacturing Laboratory (CIMLAB), scaled down module of an AS/RS is used for physical simulation purposes. This system has computer controllable miniature modules of the real AS/RSs used in production environment. The general view of the system is shown in Figure 1.1.

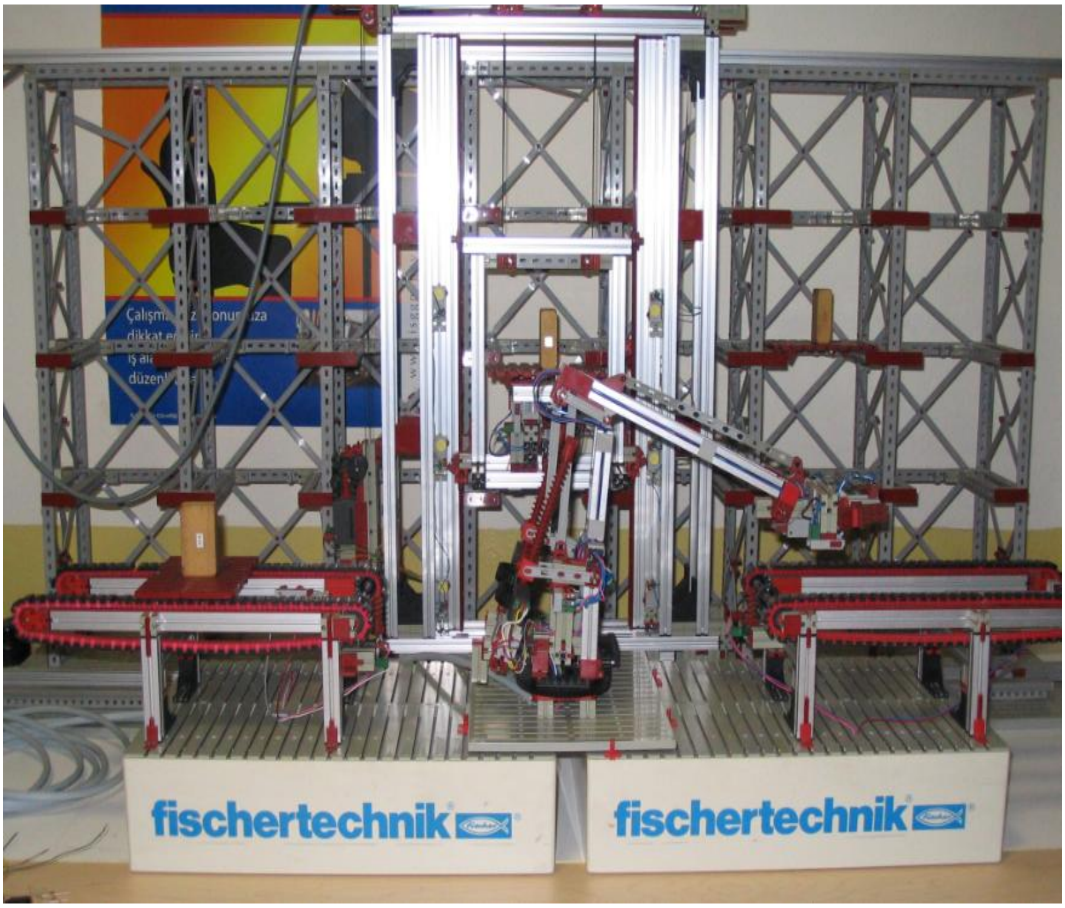


Figure 1. 1: General view of the physical system in CIMLAB

Kayalıgil et al. (1996) indicated that physical simulation has some advantages over analyses implemented on real production environment:

- Layout of the system can be changed easily.
- Extreme arrangements can be applied.
- All effects of the modifications on the system can be watched at the same time.

However, physical simulation also has some benefits when compared to computer-based simulation.

- Stochastic events are important part of the physical systems.
- Analyses made on the physical systems are realized with much ease.
- Results provided by physical simulation can be transferred to real systems with much ease.

1.1. Motivation and Scope

The main objective for this thesis to develop an experimental AS/RS test bed which can be used by researchers and students to formulate, validate and evaluate new strategies in AS/RS control. In order to build flexible test bed allowing to add new machines to the system and configuring parts of the system, a robust and flexible control system is needed. The control system comprised hardware control and software control. The hardware and its firmware provide low level control for the physical system which consists of actuators and sensors. The control software is the brain of the overall system so it must be designed considering flexibility. Following criteria are the requirements for the development of the AS/RS control system:

- System should be robust in terms of adding new equipments and configuring parts of the physical system.
- It should be possible to develop various experimental settings without changing the main code of the software.
- Parts of the system should work simultaneously and safely.
- In order to measure the performance of experiments necessary data must be calculated and recorded.
- A graphical user interface should be developed that enables using the system with ease.

For developing the control system of the AS/RS following steps are tracked:

- A survey on AS/RS related laboratories and determination of objectives
- Literature survey and determination of control rules related AS/RS
- Determination of controller architecture, development of hardware and firmware
- Development of AS/RS control software
- Integration and system tests

1.2. AS/RS Related Laboratory Facilities

AS/RSs are within the research interests of mechanical and industrial engineering so these systems are widely used in the computer integrated manufacturing systems and flexible manufacturing systems laboratories of universities for educational and test purposes. The information with respect to AS/RS related laboratory facilities around the world and Turkey will be presented.

1.2.1. Flexible Automation and Intelligent Manufacturing Systems Laboratory - BUFAIM (Boğaziçi University)

AS/RS used in this laboratory is a part of the flexible model factory. System has single rack with 36 storage cells arranged in 6X6 array, a Cartesian robot and a controller unit.(BUFAIM) Features of the system are aisle captive, single shuttle, unit load and stationary single deep rack. The AS/RS is a product of Intelitek and depicted in Figure 1.2.

Figure 1. 2: Intelitek AS/RS: Physical System and Controller Unit

Material transfer between AS/RS and other parts of the model factory performed by an Automated Guided Vehicle (AGV) with multiple load capacity and products are transferred via pallets. Cartesian robot loads pallets to storage cells and AGV and unloads pallets from storage cells and AGV.

Controller unit has a NEC V853 RISC 32-bit microcontroller and communicates with PC by a USB type cable. It has 2 integrated RS-232 channels for a teach pendant and an I/O card. Controller allows PID and PWM type of control and also has 10 speed settings. With 8 digital inputs, 4 analog inputs, 8 digital outputs and 2 analog outputs, the system can be extended if necessary. Position can be defined in (X,Y,Z) coordinates, joint coordinates, encoder counts and absolute and relative positions. (Intelitek)

Some universities around the world and Turkey use this experimental set-up. This Experimental set-up may have different configurations and features. Generally, closed loop conveyors are used for material transfer between AS/RS and other parts instead of the AGV. If AS/RS has barcode scanning system, real time control is applicable. For controlling the system, controller unit and AS/RS software module in the OpenCIM software are used by some universities as Southern Illinois University Edwardsville, Utah Valley University, Morgan State University and so forth. But in this laboratory software that controls the AS/RS is developed by researcher teams. In Turkey, Galatasaray University, Bahçeşehir University and Kırıkkale University use Intelitek AS/RS.

1.2.2. Advanced Manufacturing Technology Laboratory – AMT (Waterford Institute of Technology)

According to O’Shea (2007), AS/RS used in this laboratory is a part of the flexible model factory and its physical model, control hardware and software developed by AMT research groups. The system consists of single rack with 68 storage cells, a stacker, two conveyors and a controller unit. Features of the system are aisle captive, single shuttle, unit load and stationary single deep rack. Components of the system are shown in Figure 1.3.

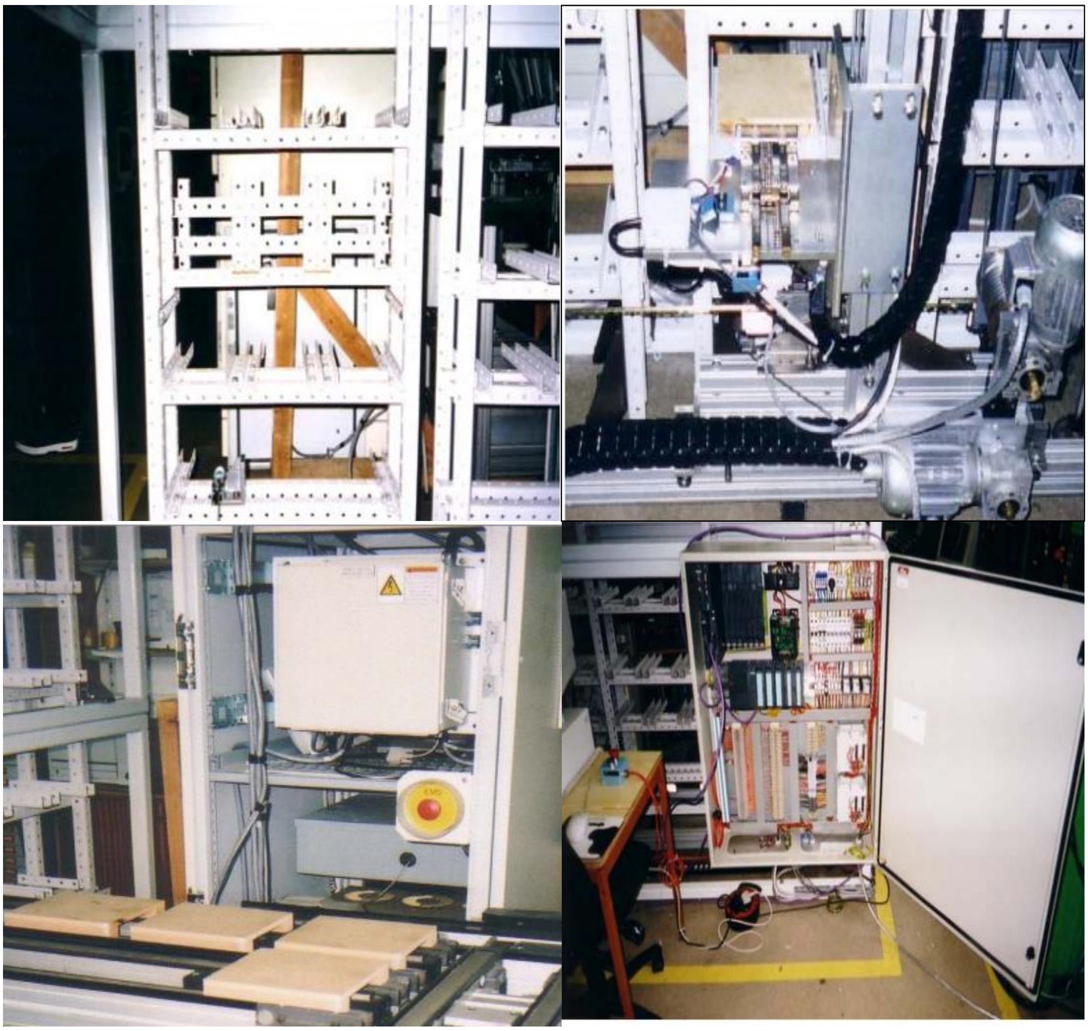


Figure 1. 3: Components of AS/RS at the AMT: Storage cells, Stacker, Input and Output Conveyors and Controller Unit

Material transfer between AS/RS and other parts of the model factory performed by input and output conveyors and products are transferred via pallets.

Siemens S7-412-2 PLC is selected as controller unit and Siemens Step 7 is used for programming the PLC. Communication between PLC and PC is provided by Profibus DP. Profibus is a standard for industrial computer network communication for automation technology. For visualization and controlling the AS/RS, WinCC software is used. Graphical User Interface (GUI) that is developed by WinCC is presented in Figure 1.4.

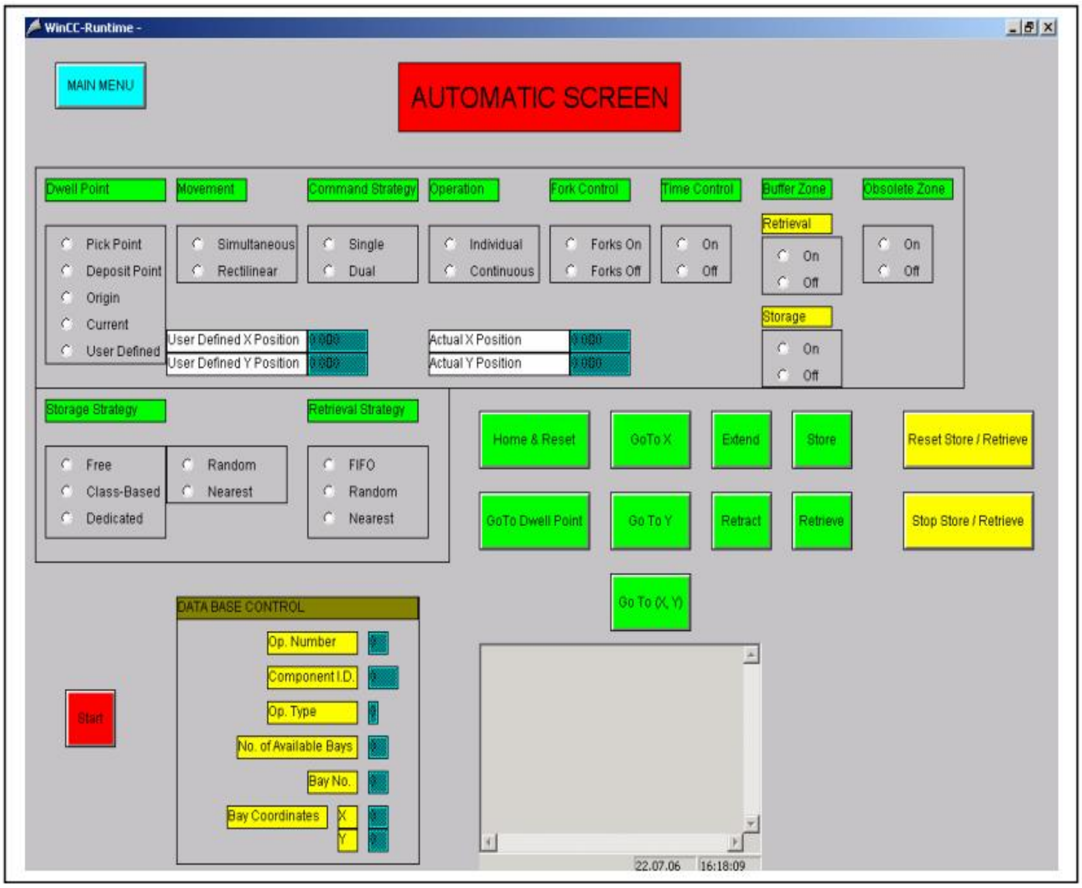


Figure 1. 4: GUI of AS/RS at the AMT

As shown in Figure 1.4, AS/RS can be controlled as automated and semi-automated. Also many of control rules regarding AS/RS (Dwell Point selection, Storage assignment and Command strategy) are set via GUI. For the purpose of recording operation of the AS/RS, MS Access database which communicates with WinCC software is used for this system.

1.2.3. University of Engineering & Technology Lahore

Farrukh et al. (2009) developed an aisle captive, single shuttle, unit load AS/RS with stationary single deep rack. Also they developed control hardware and control software of the AS/RS. General view of the AS/RS is shown in Figure 1.5.

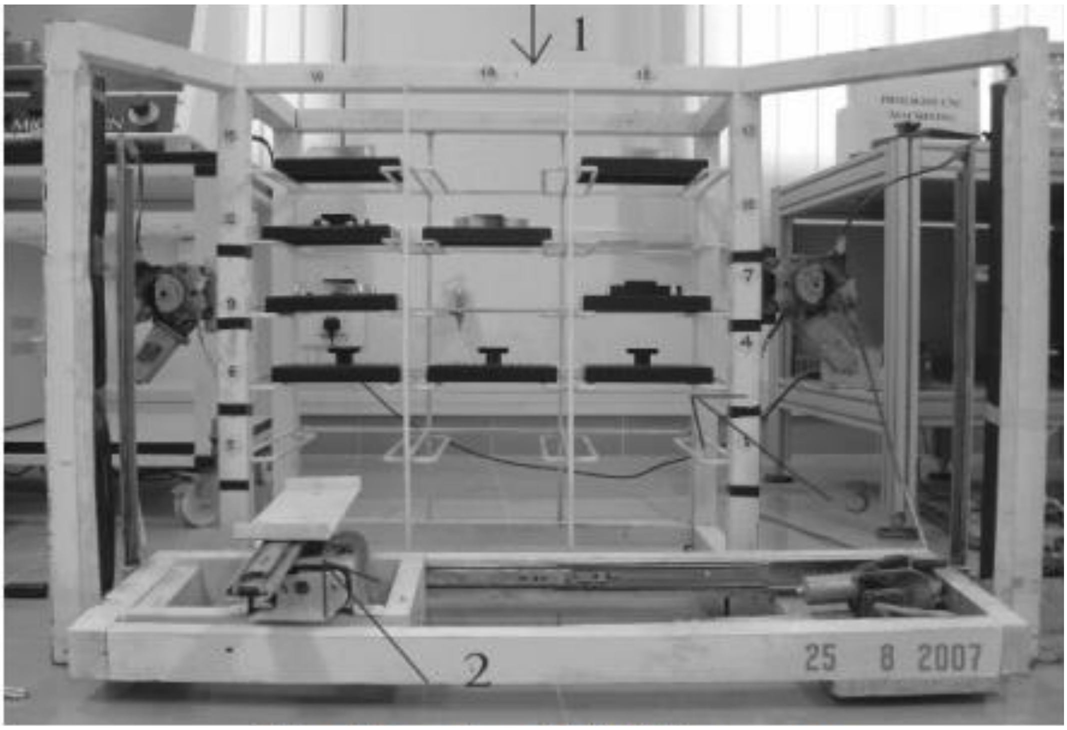


Figure 1. 5: General view of the AS/RS

System can be controlled by microcontroller and PC. In the first type control, there is a key pad for entering number that sends stacker to the storage cells and a 5X5 matrix for displaying status of the storage cells. These input and output modules and the AS/RS controlled with a microcontroller. In the second type control, software that developed in MATLAB is used and microcontroller interfaces with MATLAB and the AS/RS. Microcontroller and PC communicate via parallel port. The GUI of the control software is depicted in Figure 1.6.

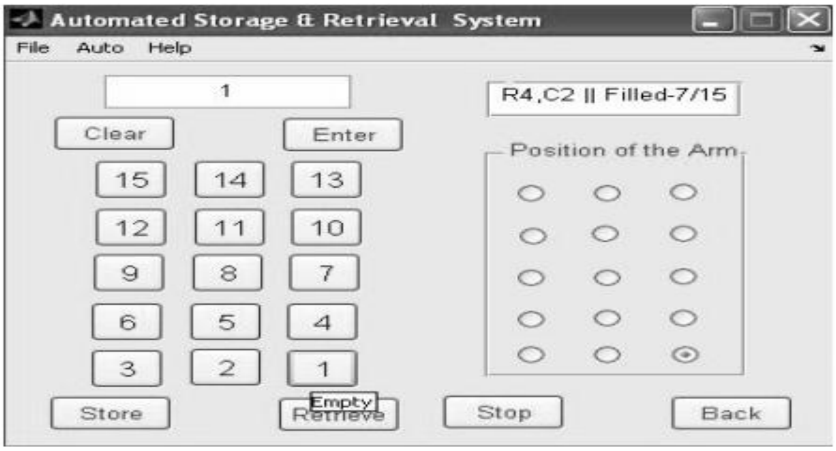


Figure 1. 6: GUI of the AS/RS

As seen in Figure 1.6, system can be used only in semi automated way. An operator enters the number that sends stacker to the storage cells for storing and retrieving purposes. Also it can display status of the storage cells.

1.2.4. Results of the Survey

Although there is in-house developed controller unit and software for AS/RS, many of the university laboratories and research centers use a product from market. Using a product from market has some disadvantages such as low flexibility, cost, limited firmware accessibility and maintainability. Typically, these products are imported so this increases the cost. Modifications on the system could be needed so it is necessary to access the firmware. But firms usually do not allow users to access the firmware. Eventually, developing boards, its firmware and software according to your requirements is suggested instead of buying a controller unit and the software for an AS/RS from the market.

1.3. Outline

Chapter 2 provides extensive background related types of AS/RS, physical design problem, control design problem which mainly include storage assignment, batching, dwell-point location and sequencing of storage and retrieval orders and performance measurement of AS/RS.

Chapter 3 presents the definition of current system which is based on the Computer Integrated Manufacturing Laboratory (CIMLAB) located in Department of Mechanical Engineering, Middle East Technical University and mechatronic design stages of this system.

Chapter 4 explains control software in details and presents graphical user interface for the system.

Chapter 5 is dedicated to case study. In this chapter, a few control policies are implemented to the system and results of the experiment are presented.

Chapter 6 presents concluding remarks and recommendations for future work regarding with the developed system.

CHAPTER 2

LITERATURE SURVEY

The Material Handling Industry of America defines an AS/RS as “An Automated Storage and Retrieval System (AS/RS) is a combination of equipment and controls that handle store and retrieve materials as needed with precision, accuracy and speed under a defined degree of automation.” In general, AS/RSs consist of stackers, aisles, racks and I/O points. Products are transported to or from racks by stackers that move through aisles between the racks. Figure 2.1 presents the general view of an AS/RS.

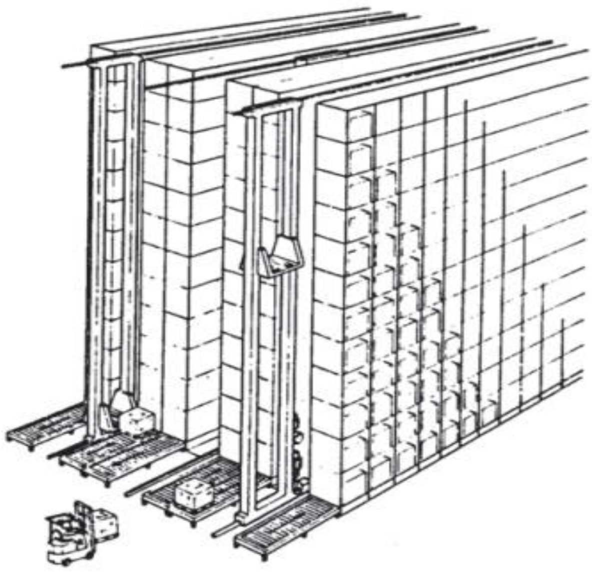


Figure 2. 1: General view of an AS/RS (Adapted from “Warehouse Modernization and Layout Planning Guide” (1985))

This section provides a summary of AS/RS options, AS/RS design problem and performance measurement of these systems. Roodbergen et al. (2008), classifies the AS/RS design problem as physical design and control design, which mainly include storage assignment, batching, dwell-point location and sequencing of storage and retrieval orders. In discussing AS/RS design problem, we take a similar approach to that of Roodbergen et al. (2008)

2.1. Automated Storage and Retrieval System Options

Roodbergen et al. (2008) classifies AS/RS options according to operational properties of AS/RS components. This classification is presented in Figure 2.2.

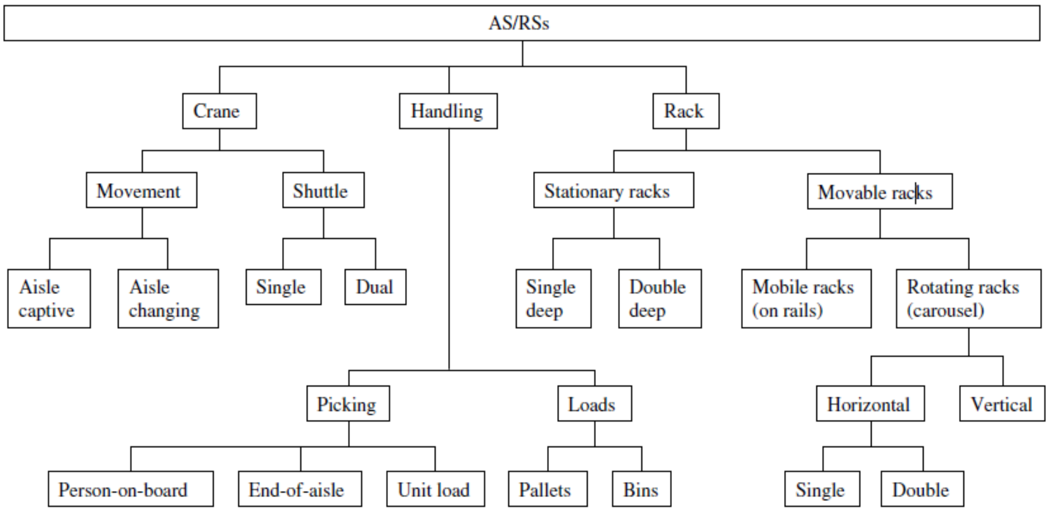


Figure 2. 2: Classification of AS/RS options (Adapted from Roodbergen et al. (2008))

Rack: The function of the racks in an AS/RS is to accommodate products. Products are stored to racks on pallets. Racks are typically metal structures. The most used racks are stationary racks but also there exists movable racks. Stationary racks have fixed location and stackers move for storing products to racks or retrieving products from racks. They have two common types as single deep and double deep. In double deep stationary racks each location allows two unit-loads to store in depth. According to Tompkins et al. (2003), the double deep racks are efficient, if the demand frequencies of products are high and the varieties of the products are low. There are two classes of movable racks which are carousels to have the capability of horizontal (Figure 2.3) or vertical movement. Horizontal carousels also have single and double deep type.

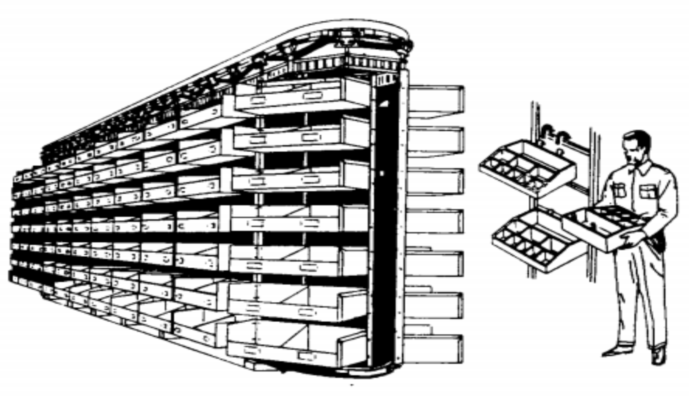
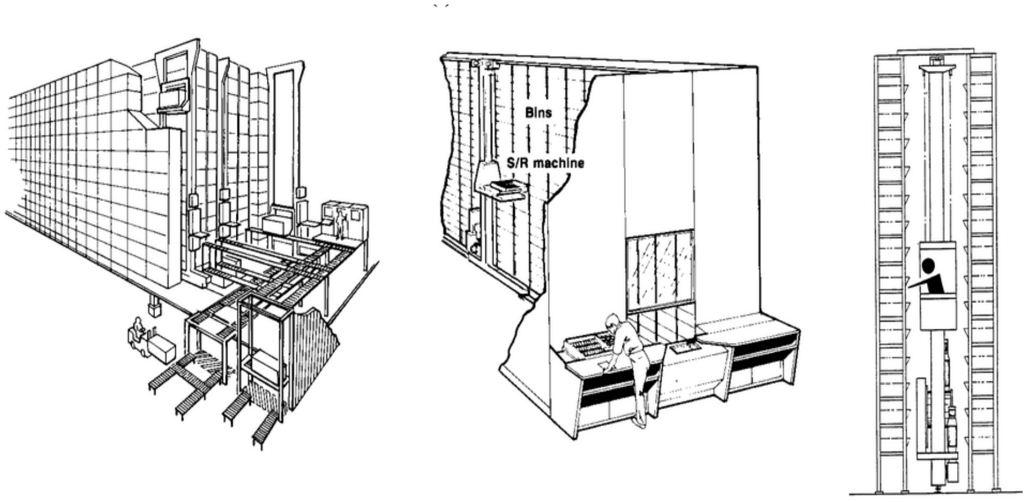


Figure 2. 3: Horizontal carousel (Adapted from "Warehouse Modernization and Layout Planning Guide" (1985))

Stackers: Stackers are designed to transport products among I/O points and racks, and also it can load products to and unload products from racks, conveyors or other parts of warehouse (e.g. automated guided vehicles). A typical stacker can perform horizontal and vertical movement simultaneously. Stackers are classified considering its movement and how many products are transported at once. Aisle captive stackers move on one aisle and do not leave this aisle but aisle changing stackers can work on more than one aisle. Stackers that can transport one product named as single shuttle stackers. Dual shuttle stackers are capable of transporting two products at the same time. Advantage of this stacker is retrieving product from rack and storing product to the rack without going to the I/O point.

Handling: Handling is separated as picking and loads. There are three picking options and the most commonly used one is the unit load (Figure 2.5.a). Unit load can be considered as a container which combines an individual material or materials and it can be moved easily by pallets. Some stackers are, referred to as person on board, designed for riding on it. For this type of picking an operator who is on the stacker, picks ordered items from racks instead of a unit load (Figure 2.5.b). The operator picks required items from unit-load at workstation and remaining of unit load is moved back to its location by stacker, this system is named as end of aisle system (Figure 2.5.c). Loads are moved in two ways with pallets and bins. Pallets are used for moving the relatively larger unit loads easily. Bins are small containers used to store products and these containers are moved without pallets.



(a) (b) (c)

Figure 2. 4: Picking options (Adapted from “Material Handling Industry of America”) (a)Unit load, (b) End of aisle, (c) Person on board

2.2. Automated Storage and Retrieval Systems Design Problem

In general, AS/RSs are used in production and distribution area. Design decisions should be decided considering in which area the AS/RS will be used. Each environment has its own priorities and features. For example, in production area manufacturing continuity is vitally important and production should not wait for the AS/RS. As for the distribution area, it is crucial to customers getting demands on time. In addition, AS/RSs are affected by other systems performance in warehouse so this point should be considered when deciding on design parameters.

Five aspects of the AS/RS design problem will be examined as physical design, storage assignment, batching, dwell point and sequencing of storage and retrieval orders.

2.2.1. Physical Design

According to Roodbergen et al. (2008), physical system must be designed considering AS/RS options and system configuration. After the selection of system options, system is configured by deciding on number of aisles, length of aisles, height of the storage racks, number of stackers per aisle, height of storage racks, equally sized or modular storage locations, number and location of I/O points, buffer capacity at the I/O points. In addition to these design decisions, it is important to pay attention to investment cost and providing future capacity requirement of company.

There are two main approaches for designing AS/RS: analytical and simulation. In the analytical methods, Bozer and White (1984) studied end of aisle system based on unit load AS/RS. They reduced pick time by modifying I/O points so it can be allocated more aisles to each picker. Chang et al. (1997) investigated the rack configurations and operating policies. An AS/RS operates in a single command cycle and dual command cycle. In a single cycle command, storage or retrieval orders are handled on a single trip. If stacker handles storage and retrieval order on a single trip, the operation is named as dual command cycle. They demonstrated that optimal rack configuration is square in time, time to reach the highest row and time to reach the farthest column are the same, for a single command cycle. But square in time rack configuration is not effective for dual cycle command. Lee et al. (2005) proposed an AS/RS type with the racks of modular size. Although many authors studied the AS/RS with equally sized racks, they developed a mathematical model for cell size. This model provides flexibility to the system.

There are a few simulation methods in the literature about physical design of AS/RS. Potrc et al.(2004) presented heuristics travel time models for AS/RS with equal-sized cells in height and randomized storage, the empty storage locations in racks have equal probability for assigning a pallet loads, under single and multi shuttle system. They demonstrated high improvement provided by using multi shuttle system instead of single shuttle system. Also, this model provides relationship between average travel times and throughput capacity considering rack types and velocity profiles of stackers.

2.2.2. Storage Assignment

Storage assignment policies are defined as allocating storage location for products in racks. There are four frequently used storage assignment policies in literature.

Random Storage Assignment: The empty storage locations in rack have equal probability for assigning a product. Although space requirement of random storage assignment is low, this policy may not be effective because of slow moving products may be located around I/O point.

Closest Open Location Storage Assignment: In this policy, pallet loads are assigned to open location where closest to the I/O point. Therefore, storage location around the I/O point is often full. This storage assignment is not effective due to assigning the pallet loads to racks regardless of its turnover.

Dedicated Storage Assignment: Each product type has a fixed storage location. It seems as an advantage assigning fast moving products to the location where closest to the I/O points but if a product is out of the stock, the dedicated location will be empty. Drawbacks of this policy are, low space utilization and high space requirements.

Class Based Storage Assignment: Racks are divided into a number of classes and the items are grouped according to demand frequency. Then, the items are assigned to own class. Random storage assignment is applied within each class. Since this policy takes into account item's demand frequency, system efficiency is high. The other advantage is class based storage policy has the flexibility of random storage policy. Three main decisions should be decided while applying class based storage policy such as where to position each of the class, how many classes the rack contains, how many items to be assigned to each class.

There are analytical and simulation studies that compare storage assignment policies in literature. Hausman et al. (1976) compare three storage assignments such as random assignment, full turnover-based assignment and class based assignment using analytical methods. Full turnover-based storage policy divides rack considering item's turnover frequency so the turnover frequencies need to be known before. This study shows that the full turnover-based storage provides more travel time reductions for stacker than other storage assignments. Need for an appropriate criterion arises while applying full turnover based storage assignment, class based storage assignment or dedicated storage assignment. Heskett (1963, 1964) presents Cube-Per-Order Index (COI) as the ratio of the item's required storage space to turnover rate.

As for simulation method, Schwarz et al. (1978) examined AS/RS in terms of storage assignment and sequencing storage and retrieval orders under stochastic conditions. Simulation showed that performance of random storage assignment and closest open location storage assignment are equivalent. Also it is demonstrated that full turnover storage assignment provides higher throughput performance than random storage assignment. Kulturel et al. (1999) investigated performance of full turnover-based storage policy and duration of stay-based policy. Goetschalckx et al. (1990) presented the duration of stay-based policy that items with the shortest duration in the storage location are assigned to the closest area to the I/O point. They used stacker's travel time as a performance criterion. Their study shows that full turnover-based storage policy outperforms duration of stay-based policy but some conditions performance difference of the full turnover-based storage policy and duration of stay-based policy is not important.

2.2.3. Batching

Orders are picked in two ways by stackers: single order picking and order batching. Stacker retrieves only one order in a single trip in the first way. The single order-picking is implemented when the stacker could not collect multiple orders at a time, or the order size is large. As for order batching, stacker collects multiple orders in a single trip. Advantage of the batching is reduction of the stacker's travel time.

There are several heuristics for batching problem and the most used is seed algorithm which is presented by Elsayed (1981). This method involves three phases such as seed selection, order addition and stopping condition. In the first step, the batches begin with selecting an order that named as seed. Seeds are selected according to several rules such as orders with smallest or largest number of items, orders with smallest or largest number of storage locations that stacker have to visit for completing the order and so forth. Then, unassigned orders are allocated to batches according to order addition rules. A variety of order addition rules available in the literature. Elsayed and Stern (1983) present order addition with the largest number of picking locations in common with the batches. De Koster et al. (1999) suggest that adding order to batches considering maximum travel time reduction. Finally, decision of batches is completed or not is determined by capacity constraints or time constraints.

Pan et al. (1995) compared the order batching algorithms through computer simulation by considering four seed selection rules and four order addition rules. They concluded that capacity of stacker is the only factor that effects the order batching algorithms. The algorithm, introduced by Hwang et al.(1988), which processes orders according to their similarity coefficients has better performance than the other 15 algorithms.

2.2.4. Dwell Point

Dwell point is a location that idle stacker is waiting for the next order. It should be decided where to locate idle stacker for minimizing the next travel time. Bozer and White (1984) presented four static dwell-point location rules for AS/RS as follows.

- Stacker is located at the input station following completion of single or dual command cycle.
- Stacker is located at the midpoint of the rack following completion of single or dual command cycle.
- Stacker is located at the input station following completion of single command cycle for storage or at the output station following completion of either single or dual command cycle for retrieval.
- Stacker remains at the last location following completion of single command cycle for storage or located at the output station following completion of either single or dual command cycle for retrieval.

Egbelu (1991) suggested dynamic dwell point selection rule which uses linear programming method. Objective function of the problem is minimizing the expected travel time of the stacker. It assumed that the frequencies of transactions are known. Egbelu and Wu (1993) compared this dynamic rule and aforementioned four static rules under random and dedicated storage assignment policies and they demonstrated that dynamic dwell point rule outperforms four static rules.

2.2.5. Sequencing of Storage and Retrieval Orders

In an AS/RS, it is important that sequencing storage and retrieval orders for efficiency and urgency. Generally, first come first served rule is implemented for storage orders. However, it should not be considered only first come first served rule for retrieval orders. It is crucial that demands to be delivered to customer before due date. While sequencing retrieval orders, system efficiency must be considered as much as due dates. Stackers operate in two ways such as single cycle command and dual cycle command. In a single cycle command, storage or retrieval orders are handled in a single cycle. If stacker performs storage and retrieval order on one trip, the operation is named as dual command cycle. Storage and retrieval orders can be paired for using dual command cycle. Considering the travel times, Graves et al. (1977) showed that dual command cycle is more advantageous than single command cycle. They mentioned single and dual command cycle as No Interleaving (NIL) and Interleaving in their study. Eben-Chaime et al. (1997) investigated the AS/RS containing several stackers under various operations management tactics. Except for high throughput levels, hybrid mode which stacker operates under single and dual command cycle outperforms dual command cycle. By decreasing travel time of stacker, system throughput can be increased and also operating cost of stacker can be reduced.

Han et al. (1987) present two methods for sequencing problem in a dynamic environment. In the first procedure, named as block sequencing, the most urgent storage and retrieval orders are selected as a block and then orders in this block are sequenced. After completed this block, new block is selected and so forth. The other method is referred as dynamic sequencing. Orders are resequenced considering due dates when new order is received. In literature there are various methods for dynamic sequencing. Some of these methods are summarized as follows.

First Come First Served: Retrieval orders are chosen in order of arrival time.

Shortest Completion Time: Retrieval order which has shortest completion time is handled first.

Nearest Neighbour: Han et al. (1987) suggested this method. In one cycle a storage and a retrieval orders are selected such that the distance between storage and retrieval location is minimum.

Shortest Leg: Han et al. (1987) also suggested this method. In this method, storage locations and retrieval locations are selected such that sum of the travel time of storage location and the travel time between storage and retrieval location is minimum.

Han et al. (1987) compare nearest neighbour method with first come first served method and they concluded that nearest neighbour method has lower average travel time. They also compare nearest neighbour method with shortest leg method. At the end of the long run nearest neighbour method is better than shortest leg method. Because, pallet loads are assigned to close locations to the I/O point and locations far from the I/O points remain open. Therefore, this situation increases the travel time of stacker.

2.2.6. Performance Measurement of Automated Storage and Retrieval Systems

There are several performance criteria in literature for evaluating design and control rules of an AS/RS. Most of the researchers select travel time as performance measure for an AS/RS. Otherwise, waiting times of products, waiting times of stackers, number of products handled per time period and number of products waiting in queue is studied as performance criteria.

Hausman et al. (1976) developed travel time models for unit load AS/RSs which operate single command cycle. In their model I/O point is at the end of the rack and they investigated random, full turnover and class based storage assignments. Graves et al. (1977) studied storage assignment and interleaving policies. Bozer et al. (1984) present travel time models for alternative I/O points by applying dual command cycle and random storage assignment. Also, they examined various dwell point strategies. Chang et al. (1995) took into account the speed profile of stacker with known acceleration and deceleration under random storage policy. Chang et al. (1997) extend their work by including the rack configuration. Wen et al. (2001) studied class based storage assignment and full turnover based storage assignment with various speed profiles of stacker.

2.3. Results of the Survey

As can be observed throughout this chapter, the AS/RS problems are interrelated with each other. Figure 2.6, which has been adapted from Roodbergen et al.(2008), summarizes the relationships between decisions of AS/RSs.

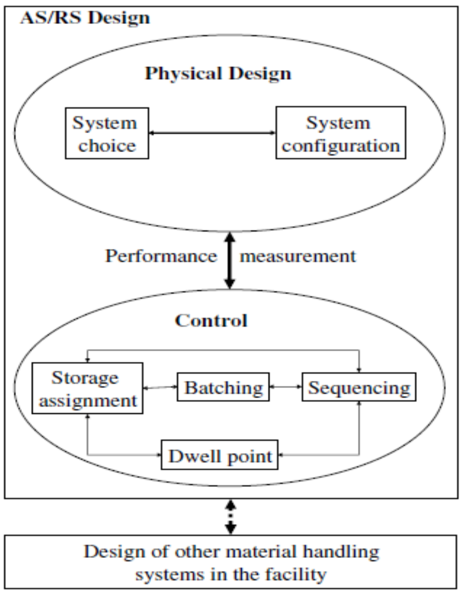


Figure 2. 5: A summary of the AS/RS design problems (Adapted from Roodbergen et al. (2008))

As it is seen from Figure 2.6, physical design and control design decisions affect each other and consequently affect the whole system performance. Also an AS/RS is a part of the warehouse so other equipments (AGVs, conveyors and so forth) of warehouse should be considered when deciding on AS/RS design decisions. For example, the other system should be taken into account when deciding number of I/O points and its locations. Due to the investment cost of the AS/RSs high, it is important to consider all of the parameters when designing the system. Furthermore, system should satisfy current and future demand requirements.

When the literature on the AS/RS design problems is observed in general, most of the studies have been done in static environment. Whereas, today's competitive business environment should satisfy changing customers' demand. Dynamic solution techniques of AS/RS design problems provide better results than static techniques.

Another lack of the literature is the fact that the studies consider a limited number of design decisions simultaneously. So we cannot obtain global optimum solutions and this causes differences between theoretical studies and applications in practice.

CHAPTER 3

MECHATRONIC DESIGN

Mechatronic systems are integration of physical systems that includes sensors and actuators, electronics, computers and controls. In mechatronic systems, there is a data flow between physical system and control system. Control system is the intelligence part of the overall system and the physical system is for the interaction between control system and the world. Control system might be microcontroller units or interfaced via interfacing environment to computers.

In the scope of this thesis, a customized interfacing environment as a low level control system is designed and manufactured for an AS/RS which is based on the Computer Integrated Manufacturing Laboratory (CIMLAB) located in Department of Mechanical Engineering, Middle East Technical University. This chapter includes in depth explanation of system architecture, physical miniaturized system and interfacing environment.

3.1. Architecture

A customized interfacing environment for the AS/RS is designed and manufactured as a part of this thesis work. As shown in the Figure 3.1, there will be high level control environment as desktop given a physical miniaturized system as AS/RS which is to be controlled. Also interfacing medium will be the low level control environment.

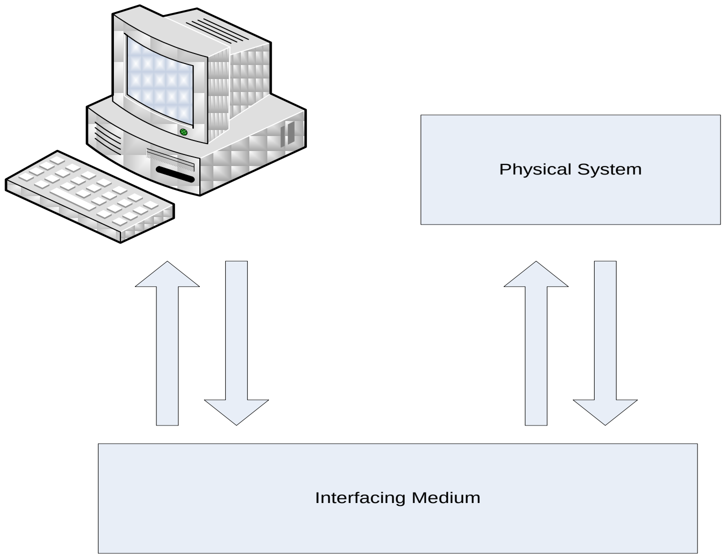


Figure 3. 1: General System

From the literature and AS/RS related laboratories survey, it is seen that the current AS/RS has main components of similar systems as conveyors, stacker, rack and a robot arm. So it is not expected too much expansion in itself. Furthermore, the current system may be a subsystem of a warehouse. At this point, two issues are important. Firstly, components of the physical system should work synchronously. The modules of the interfacing environment are selected considering this criterion. On the other hand, if the current system will be a subsystem of a warehouse or an expansion will be need for the AS/RS, the design should allow the expansion for future developments. So there will be two boards which communicate with the computer one at a time. When system demands expansion, limited number of boards can be connected to the computer and communicates with the current AS/RS over the computer (Figure 3.2). This expansion is limited by the number of computer serial ports.

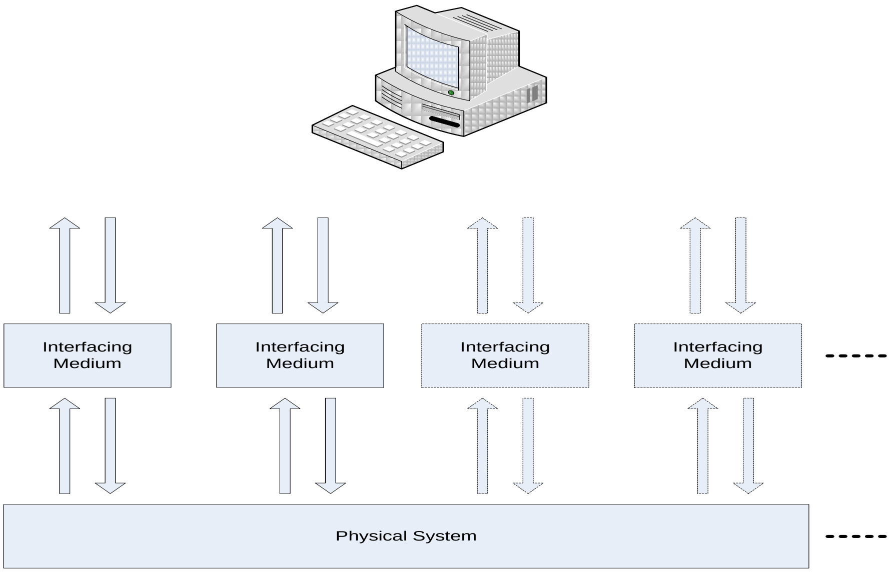


Figure 3. 2: Expanded architecture

Communication between the control program and interfacing medium can be accomplished by various communication protocols. Every communication protocol has advantages and disadvantages. Potential communication protocols will be addressed below and one of them will be selected according to ease of usability, speed and expandability as a design criterion.

Serial Port: In serial port communication, data is transferred in or out one bit at a time over cables that have RS-232 standard. 9 pin D type connector is specified for this standard. Serial communication allows synchronous or asynchronous transmission. Connection speed can be increased up to 115 Kb/s for modern computers.

Parallel Port: In parallel port communication, information transferred in or out over more than one wire. Maximum speed of the communication is 2.5 Mb/s. Parallel ports are not put on modern computers with the aim of cost saving.

USB: Serial communication protocol USB is designed for computer peripherals to share a standard communication. Depending on the version, USB has four types transfer speed as 1.5 Mb/s, 12 Mb/s, 480 Mb/s and 4000 Mb/s. USB connectors also carry 5V and 500 mA so some devices are driven by only USB cables.

Firewire: Firewire is a serial bus interface standard used by computers, digital video and so forth. It offers high speed communications and supports isochronous and asynchronous applications. It has two versions whose transfer rates are 400 Mb/s and 800 Mb/s.

The current AS/RS is a slow system and it is not expected to involve much expansion in the future in itself. All the communication alternatives are sufficient for our demands. So we make our selection considering simplicity and select serial port communication for interfacing between the computer and boards. Although serial port is seen as legacy ports for modern computers, there are several USB to RS232 converters in the market.

As we mentioned previously, it is designed two boards as an interfacing medium. The two boards installed are designated as Handling Board and Stacker Board. Although there will not need an information flow constantly, information flow is required between these two boards for system components to work properly. There are two main protocols for peripheral communication: inter integrated circuit (I2C) and serial peripheral interface (SPI). Instead of using these protocols, it is decided that the control program that runs on computer as master and boards as slave. Each board gets information which is necessary for itself from the low level control system and sends information to the control program. All the peripheral information is collected at the control program and boards communicate with each other over the control program. Figure 3.3 shows the overall architecture of the system.

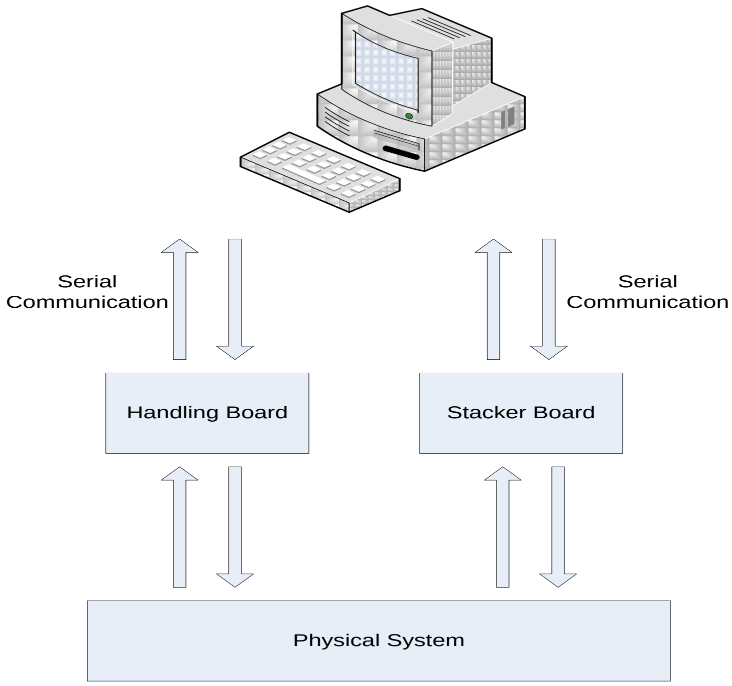


Figure 3. 3: Overall architecture

3.2. Automated Storage and Retrieval System-CIMLAB Test-Bed

The physical system housed in the Computer Integrated Manufacturing Laboratory (CIMLAB) mainly consists of six parts. Components of the AS/RS are presented in detail in terms of tasks in the system and their properties. Figure 3.4 shows general view of the system.

Figure 3. 4: Layout of the system

Input Pallet: It is a bidirectional and constant speed conveyor. A 12V DC motor drives the conveyor belt transmitting through a gearbox. There are two limit switches at both ends of the conveyor for stopping the pallet. A pallet is fixed on the conveyor. Speed of the conveyor is 4 cm/s and length of it is 40 cm. It can transport only one product in one cycle. The control is performed by Handling Board.

Task of the input pallet is moving the incoming products towards the robot arm. When a product enters the system, it is assumed to be located on the pallet by an operator. Then, the pallet moves towards the robot while the limit switch, which is on the robot arm side, is off. After the limit switch is on, the pallet stops and waits for unloading by the robot arm. When the robot arm unloads the pallet, it moves in the reverse direction for transporting a new storage order. The pallet stops when the other limit switch is on and so forth.

Robot Arm: It is a four degree of freedom serial manipulator with a length of joints 10 cm, 7 cm, 15 cm. There are four 12V DC motors on the robot arm with same speed. Three of the motors for joint movements and the other one is for gripping products. Three optical encoder sensors are used for sensing the robot arm position. Also there are five limit switches on the robot. Four of the limit switches are for referencing of the robot. At the beginning of the Handling Board firmware which controls the robot arm and the input pallet, motors move towards the reference position. If any of the limit switches is on, related motor stops. The other limit switch is for sensing that the product has been gripped or not.

The robot arm serves the input pallet, the output pallet and the stacker. After the picking up the incoming product from the input pallet, the robot arm moves towards I/O point. If the stacker is also at the I/O point, the robot deposits the product to the pallet which is on the shuttle. If the stacker brings a product from rack to the I/O point, the robot arm will pick up this product and move towards the output pallet. When the robot arm arrives to the output pallet, it loads product to the output pallet.

Output Pallet: It is a bidirectional and constant speed conveyor. A 12V DC motor drives conveyor belt transmitting through a gearbox. There are two limit switches at both ends of the conveyor for stopping the pallet. A pallet is fixed on the conveyor. Speed of the conveyor is 4 cm/s and the length of it is 40 cm. It can transport only one product in one cycle. The control is performed by Stacker Board.

Task of the output pallet is moving the outcoming products towards an operator. After unloading product from stacker, product is loaded on the pallet by the robot arm. Then pallet moves towards the operator while the limit switch, on the operator side, is off. After the limit switch is on, conveyor stops and waits for unloading by the operator. When the operator unloads the pallet, it moves in the reverse direction for transporting a new retrieval request. The pallet stops when the other limit switch is on and so forth.

Stacker: Stacker has the ability to move in three directions: horizontal, vertical and shuttle movement. There are three 12V DC motors on the stacker for moving the machine in three directions. Two DC motors, which provide horizontal and vertical movement, are more powerful than the shuttle motor. The horizontal, vertical and shuttle motors have a speed of 3 cm/s, 1 cm/s, 5 cm/s, respectively. Six limit switches are used for sensing the stacker is at which row. In addition, the column the stacker at is sensed by seven limit switches on the aisle. The other two limit switches are used at the bounds of the shuttle for locating products to racks or picking up products from racks. The control is performed by the Stacker Board.

The main task of the stacker is storing products to the racks and retrieving products from the rack. Before the robot arm carries the product to the I/O point, stacker takes the idle pallet from fourth column. The fourth column in the rack is devoted to idle pallets. Then, stacker with empty pallet moves towards I/O point for taking the product. After the stacker is loaded, it moves to a storage location assigned by the control program. When the control program sends retrieval order to the AS/RS, stacker moves to retrieval location chosen again by the control program. Then, stacker picks up the product from rack by the shuttle and returns to its I/O point. After the robot arm picks up the product from the stacker, stacker moves to the fourth column to return the idle pallet to the rack for the next storage order. If there is not an immediate storage order, the stacker remains at last location or moves to selected Dwell-Point according to the selected control rules.

Aisle: Aisle can be considered as a rail for the stacker moves easily on. Length of the aisle is 2m. There are seven limit switches on the aisle for sensing each column. Limit switches transmit this information to the Stacker Board.

Racks: It has 21 storage cells arranged in 3X7 array. All of the cells are equal size. Dimensions of the each cell are 20cmX20cmX20cm.

Photographs of the aforementioned components of the system are presented in Appendix A.

3.3. Boards

There are four components of the AS/RS as input pallet, output pallet, robot arm and stacker which are controlled by boards. Two boards are designed for controlling these components. Handling Board controls the input pallet and the robot arm. Stacker Board controls the output pallet and the stacker. These design decisions are made considering the number of digital I/O on the components and roles of the components in the system.

3.3.1. Handling Board

This subsection will cover detailed explanation of the Handling Board's hardware, circuit design and firmware.

Hardware

Microcontroller selection is the first stage of design process. Selection is made considering the following criteria:

- Chip's resources (interrupt types, number of pins and interrupts)
- Speed of the chip
- Availability on the market
- Availability of development tools

Handling Board controls the input pallet and the robot arm. Input pallet consists of two limit switches and a DC motor. Robot arm consists of five limit switches, four DC motors and three encoders. Considering these actuators and sensors, selected microcontroller should provide at least 20 digital I/O. Reflective optical sensors have a critical importance for positioning the robot arm. If any of these 3 encoders' status changes, the embedded program must interrupt the program routine. So there will be need 3 on change interrupts for reflective optical sensors. Also there will be a need of an external interrupt for input pallet's limit switches. For communicating with computer, selected chip should have a serial communication port. It is mentioned before that speed of the system is low so most of the microcontroller's in the market satisfy the speed constraint. After eliminating the chips considering resources and features, selecting the chip brand and the specific model remain. This decision is made according to the availability in the market, the popularity of the model and the availability of development tools. Considering these criteria Microchip's 16F877A microcontroller with 40 pin is selected for Handling Board.

Input pallet consists of two limit switches and a DC motor. Limit switches are normally in "off" state. If one of the limit switches is in "on" state, motor should stop. Only one of them can be in the "on" state simultaneously. So two switches are connected to the "or" gate (74HC32) and the output of the "or" gate is connected to external interrupt (Figure 3.5).

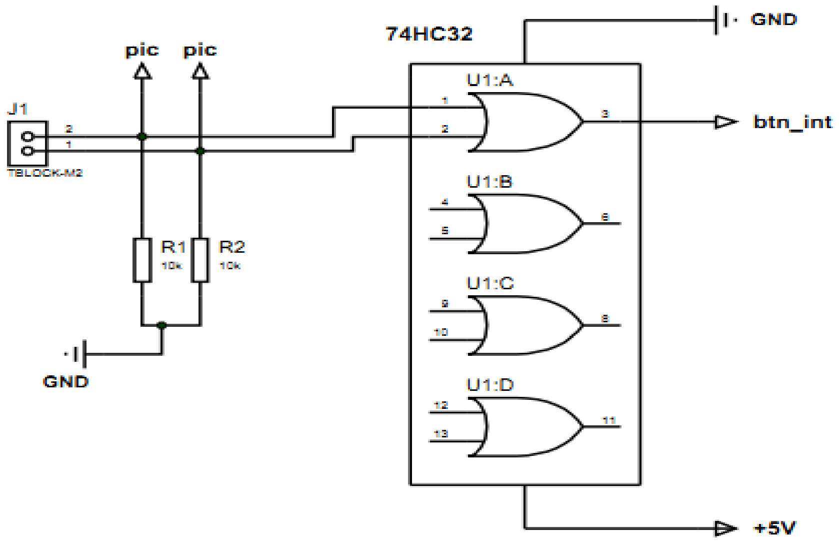


Figure 3. 5: Limit switches connection details

A dual full bridge driver L298 is used for driving the DC motor. L298 can drive two motors and operating voltage is 46V with maximum 4A current. DC motor is driven with fixed speed so PWM output is not used. Operating voltage of the DC motor is set as 12V and this supply is separated from the Handling Board circuitry. For this purpose a connector is used for an external power supply (Figure 3.6).

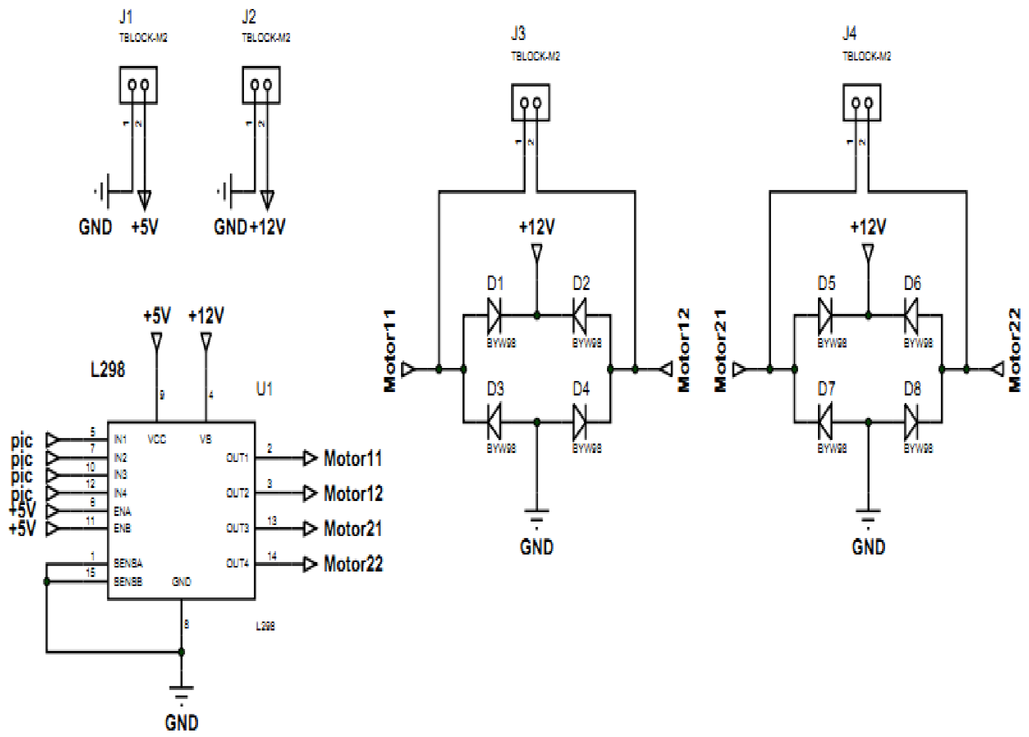


Figure 3. 6: Motor Driver Unit

Robot arm consists of five limit switches, four DC motors and three encoders and reflective optical sensors. Four of the limit switches are for referencing the robot arm. The fifth one is for detecting whether the product has been picked or not. It is not needed to use interrupt for the limit switches. So these five limit switches are directly connected to the microcontroller as digital input.

Four DC motors are driven with two L298 drivers. DC motor is driven with fixed speed so PWM output is not used. Operating voltage of the DC motors is set as 12V and this supply is separated from the Handling Board circuitry.

Three reflective optical sensors are used for detecting the position of the robot arm. Sensors are connected on change interrupts separately. Because of the analog output of the reflective optical sensor Hex Schmitt trigger inverter (74HC14) is used for detecting signal correctly (Figure 3.7).

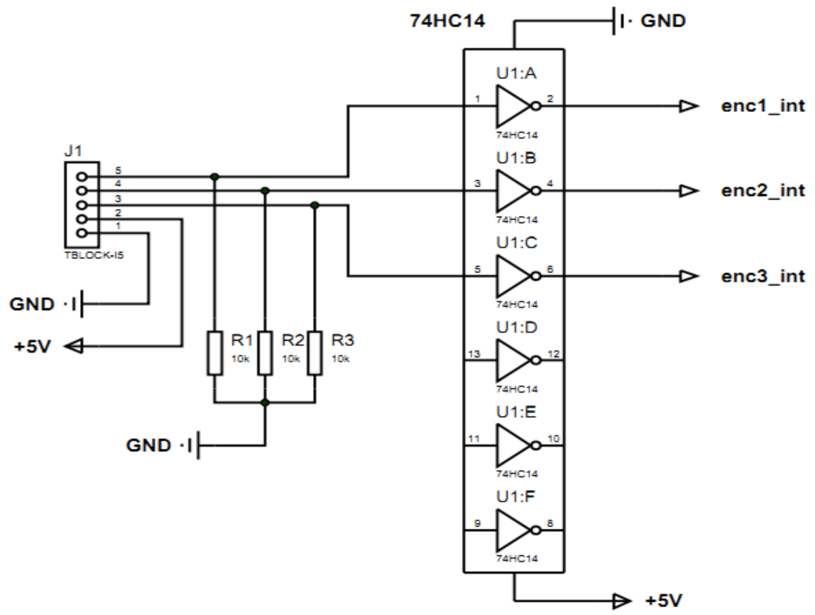


Figure 3. 7: Encoders interface

There is a RS232 communication unit for communication between the board and computer (Figure 3.8).

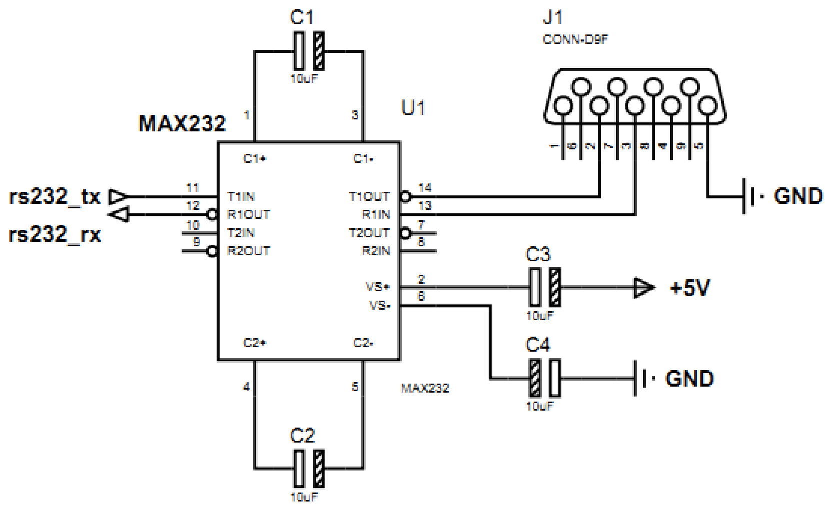


Figure 3. 8: RS232 Communication Unit

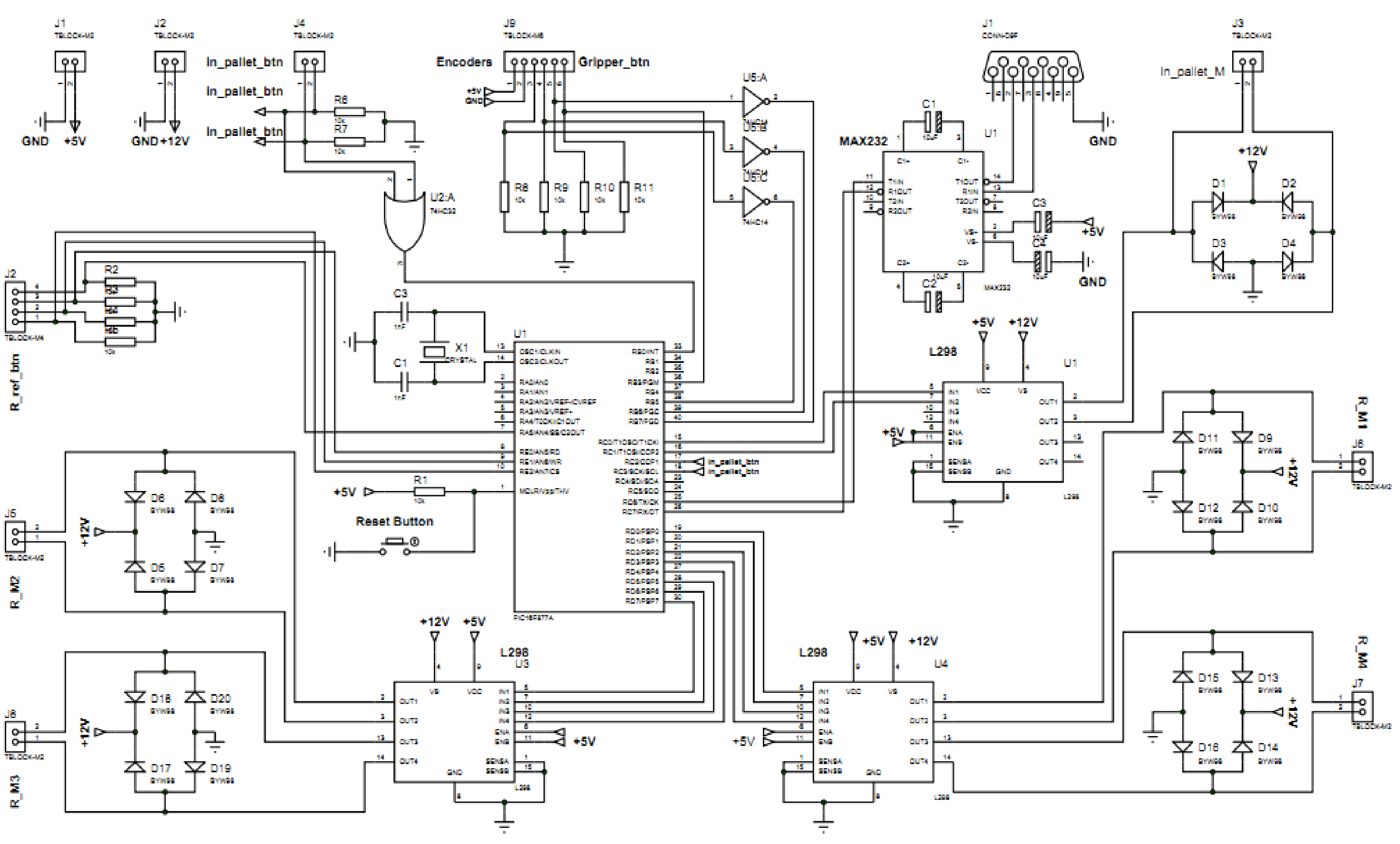


Figure 3. 9: Schematic of the Handling Board

Table 3. 1: PIC 16F877A pin usages at the Handling Board

Pin#	Connected to	Description	Pin#	Connected to	Description
1	Reset push button	Resetting system	40	74HC14 output	Encoder of Motor3
2	Unused		39	74HC14 output	Encoder of Motor2
3	Unused		38	74HC14 output	Encoder of Motor1
4	Unused		37	Unused	
5	Unused		36	Limit switch	Gripper of robot arm
6	Unused		35	Unused	
7	Limit switch	Referencing for robot arm	34	Unused	
8	Limit switch	Referencing for robot arm	33	74HC32 output	Interrupt for input pallet's switches
9	Limit switch	Referencing for robot arm	32	Supply voltage	
10	Limit switch	Referencing for robot arm	31	Ground	
11	Supply voltage		30	L298 input	Motor3 of robot arm
12	Ground		29	L298 input	Motor3 of robot arm
13	Oscillator		28	L298 input	Motor2 of robot arm
14	Oscillator		27	L298 input	Motor2 of robot arm
15	L298 input	Input pallet motor	26	Max232	RX
16	L298 input	Input pallet motor	25	Max232	TX
17	Limit switch	Input pallet	24	Unused	
18	Limit switch	Input pallet	23	Unused	
19	L298 input	Motor1 of robot arm	22	L298 input	Motor4 of robot arm
20	L298 input	Motor1 of robot arm	21	L298 input	Motor4 of robot arm

Firmware

Two types of firmware are loaded to the microcontroller. The first one is bootloader firmware and the other is application firmware. The bootloader firmware is written to the flash memory of microcontroller and it enables programmers to load their application firmware to the microcontroller via computer by serial port. The bootloader firmware is needed to be loaded to microcontroller by programmer hardware. Then application firmware is loaded to microcontroller via computer by a serial port.

The firmware starts with initialization of variables. The firmware includes three interrupt routines as on change, external and serial communication and robot arm related functions as reference, unload pallet, load pallet, load stacker and unload stacker. Then main loop starts with setting output ports as low (0 V) for stopping the motors. After completion these settings main function waits until receiving data from the serial port. If some data is received from the serial port the serial port interrupt routine starts and status variables are updated accordingly. Control program which runs on the computer do not send any data if the board state is busy. In the main loop, robot arm related functions are called according to the status variables. These functions call the position function for controlling the robot arm position. Position function decides whether the robot arm is at the desired position by polling method. The polling method uses the three encoder data for making the decision. Encoder data is read by on change interrupt routine. After completing any of the robot arm related functions, board state is set as not busy and status variable is set at six. The status variable which notifies the completion of the function is sent to the computer by serial port. The external interrupt routine controls the input pallet movement. Communication protocol between control program which runs on the computer and Handling Board is shown at Table 3.3.

Table 3. 2: Handling Board – Control Program Communication Protocol

Handling Board – PC Communication Protocol			
Received Data	Description	Send Data	Description
0	Reference	0	Unused
1	Unload Pallet	1	Unused
2	Load Stacker	2	Unused
3	Unload Stacker	3	Unused
4	Load Pallet	4	Unused
5	Unused	5	Unused
6	Unused	6	Board is ready
7	Unused	7	Unused

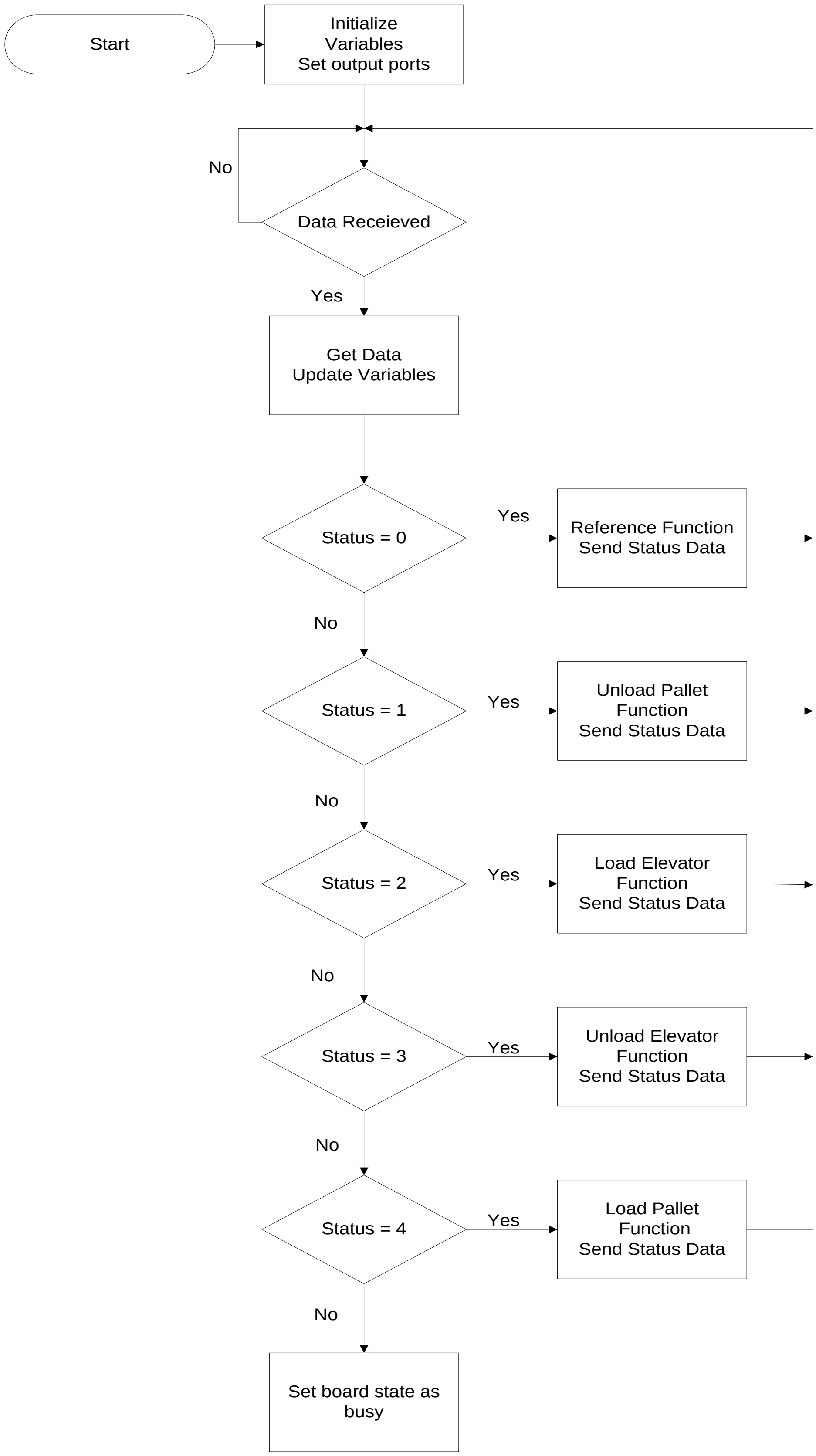


Figure 3. 10: Flowchart of the Handling Board Firmware

3.3.2. Stacker Board

This subsection will cover detailed explanation of the Stacker Board’s hardware, circuit design and firmware.

Hardware

Microcontroller selection is the first stage of design process. Selection is made considering following criteria:

- Chip’s resources (interrupt types, number of pins and interrupts)
- Speed of the chip
- Availability on the market
- Availability of development tools

Handling Board controls the output pallet and the stacker. The output pallet consists of two limit switches and a DC motor. The stacker consists of six limit switches on the vertical direction, seven limit switches on the horizontal direction, two limit switches for the shuttle movement and three DC motors. Considering these actuators and sensors, selected microcontroller should provide at least 25 digital I/O. Limit switches have a critical importance for positioning the stacker. If any of these 15 limit switches’ status change, the embedded program should interrupt the program routine. So there will be need 3 on change interrupts for limit switches. Also there will be a need of an external interrupt for output pallet’s limit switches. For communicating with computer, selected chip should have a serial communication port. It is mentioned before that speed of the system is low so most of the microcontroller’s in the market satisfy the speed constraint. After eliminating the chips considering resources and features, selecting the chip brand and the specific model remain. This decision is made according to the availability on the market, the popularity of the model and the availability of development tools. Considering these criteria Microchip’s 16F877A microcontroller with 40 pin is selected for Stacker Board.

Output pallet consists of two limit switches and a DC motor. Limit switches are normally in “off” state. If one of the limit switches is in “on” state, the motor should stop. Only one of them can be in “on” state simultaneously. So two switches are connected to the “or” gate (74HC32) and the output of the “or” gate is connected to external interrupt.

A dual full bridge driver L298 is used for driving the DC motor. The DC motor is driven with fixed speed so PWM output is not used. Operating voltage of the DC motor is set as 12V and this supply is separated from the Stacker Board circuitry.

There are seven limit switches on the floor for controlling the stacker is at which column. Only one of them can be in “on” state simultaneously. So seven switches are connected “or” gate (74HC32) and the output of the “or” gate is connected to on change interrupt. On which row the shuttle is controlled by six limit switches which are on the vertical direction. Similar to horizontal switches, one of the vertical switches can be in “on” state simultaneously and vertical switches are connected “or” gate (74HC32) and the output of the “or” gate is connected to on change interrupt. Two limit switches control the shuttle movement. One of them is at the forward limit of the shuttle and the other is at the reverse limit of the shuttle. One of the two switches can be in “on” state simultaneously. So two switches are connected “or” gate (74HC32) and the output of the “or” gate is connected to on change interrupt.

Three DC motors are driven with two L298 drivers. All of the DC motors are driven with fixed speed so PWM output is not used. Operating voltage of the DC motors is set as 12V and this supply is separated from Handling Board circuitry. For this purpose a connector is used for an external power supply. Two DC motors move stacker horizontally and vertically and the other is for the shuttle movement.

There is a RS232 communication unit and connection details are shown in Subsection 3.3.1. Schematic view of the Stacker circuitry is presented in Figure 3.11.

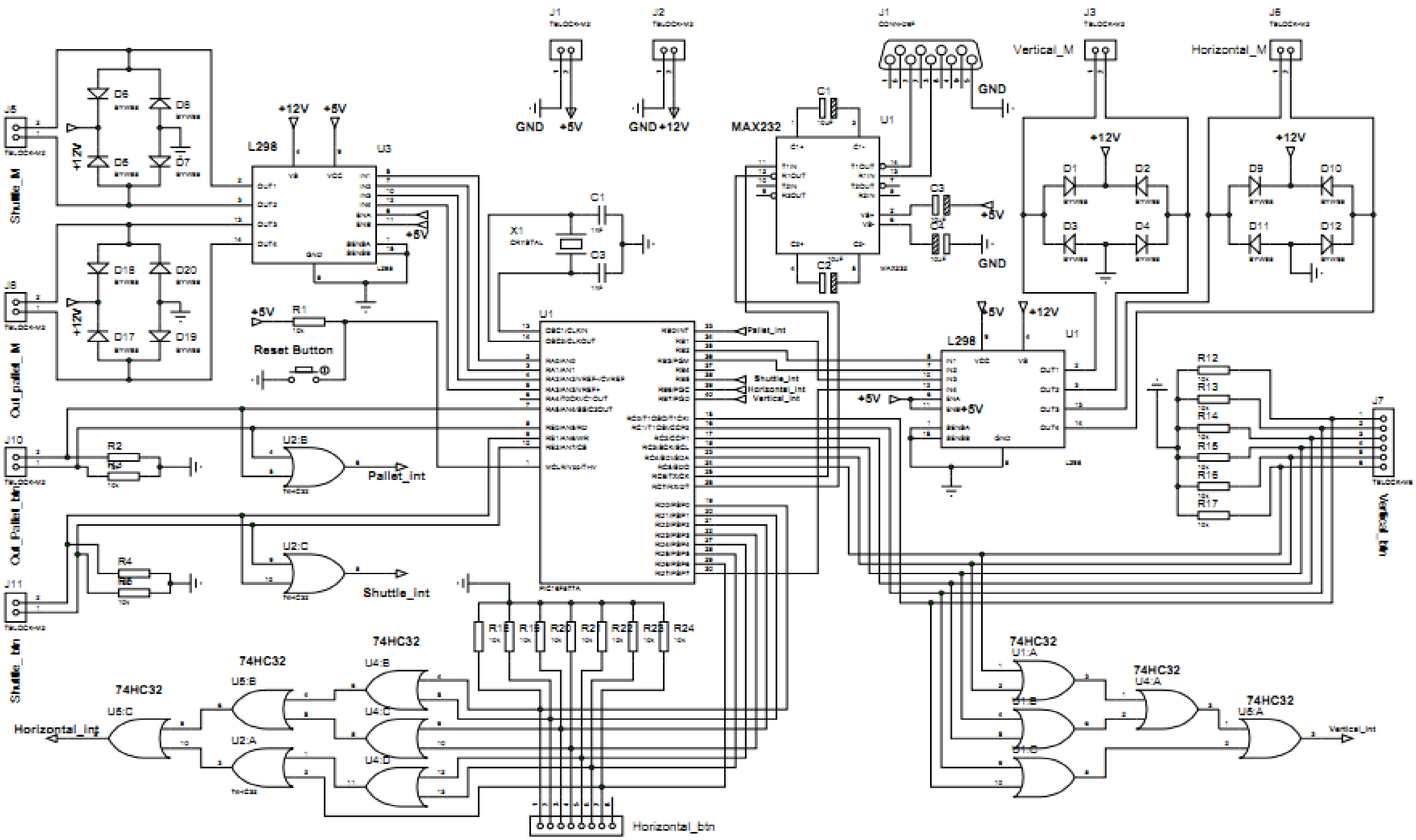


Figure 3. 11: Schematic of the Stacker Board

Table 3. 3: PIC 16F877A pin usages at the Stacker Board

Pin#	Connected to	Description	Pin#	Connected to	Description
1	Reset push button	Resetting system	40	74HC32 output	Interrupt for vertical stacker switches
2	L298 input	Shuttle motor	39	74HC32 output	Interrupt for horizontal stacker switches
3	L298 input	Shuttle motor	38	74HC32 output	Interrupt for shuttle switches
4	Limit switch	Input pallet	37	L298 input	Vertical motor of stacker
5	Limit switch	Input pallet	36	L298 input	Vertical motor of stacker
6	Unused		35	L298 input	Horizontal motor of stacker
7	L298 input	Output pallet motor	34	L298 input	Horizontal motor of stacker
8	L298 input	Output pallet motor	33	74HC32 output	Interrupt for output pallet's switches
9	Limit switch	Reverse limit of shuttle	32	Supply voltage	
10	Limit switch	Forward limit of shuttle	31	Ground	
11	Supply voltage		30	Unused	
12	Ground		29	Limit switch	Horizontal
13	Oscillator		28	Limit switch	Horizontal
14	Oscillator		27	Limit switch	Horizontal
15	Limit switch	Vertical	26	Max232	RX
16	Limit switch	Vertical	25	Max232	TX
17	Limit switch	Vertical	24	Limit switch	Vertical
18	Limit switch	Vertical	23	Limit switch	Vertical
19	Limit switch	Horizontal	22	Limit switch	Horizontal
20	Limit switch	Horizontal	21	Limit switch	Horizontal

Firmware

The firmware starts with initialization of variables. The firmware includes three interrupt routines as on change, external and serial communication and stacker related functions as reference, load shelf and unload shelf. Then main loop starts with setting output ports as low (0 V) for stopping the motors. After completion these settings main function waits until receiving data from the serial port. If some data is received from the serial port the serial port interrupt routine starts and status variables are updated accordingly. Control program which runs on the computer do not send any data if the board state is busy. The received data from control program is one byte and it covers information about function type, where to locate stacker at vertical and horizontal direction. Mod 4 of the received data gives information about which function the firmware will execute. Information of stacker location procedure is presented at Table 3.5. In the main loop, stacker related functions are called according to the status variables. These functions decide whether the stacker is at the desired position by polling method. The polling method uses the fifteen limit switches data for making the decision. Limit switches data is read by on change interrupt routine. After completing any of the stacker related functions, board state is set as not busy and status variable is set at six. The status variable which notifies the completion of the function is sent to the computer by serial port. The external interrupt routine controls the output pallet movement. Communication protocol between the control program and the Stacker Board is shown at Table 3.6.

Table 3. 4: Calculating the stacker location from received data

Mod 8 of Received Data/16	Location X	Mod 4 of Received Data/4	Location Y for load shelf function	Location Y for unload shelf function
0	Unused	0	Unused	Unused
1	1	1	1*2 = 2	1*2 - 1 = 1
2	2	2	2*2 = 4	2*2 - 1 = 3
3	3	3	3*2 = 6	3*2 - 1 = 5
4	4			
5	5			
6	6			
7	7			

Table 3. 5: Stacker Board – Control Program Communication Protocol

Mod 4 of Received Data	Description	Send Data(Bit)	Description
0	Reference	0	Unused
1	Load Shelf	1	Unused
2	Unload Shelf	2	Unused
3	Pallet Send	3	Unused
		4	Unused
		5	Unused
		6	Board is ready
		7	Unused

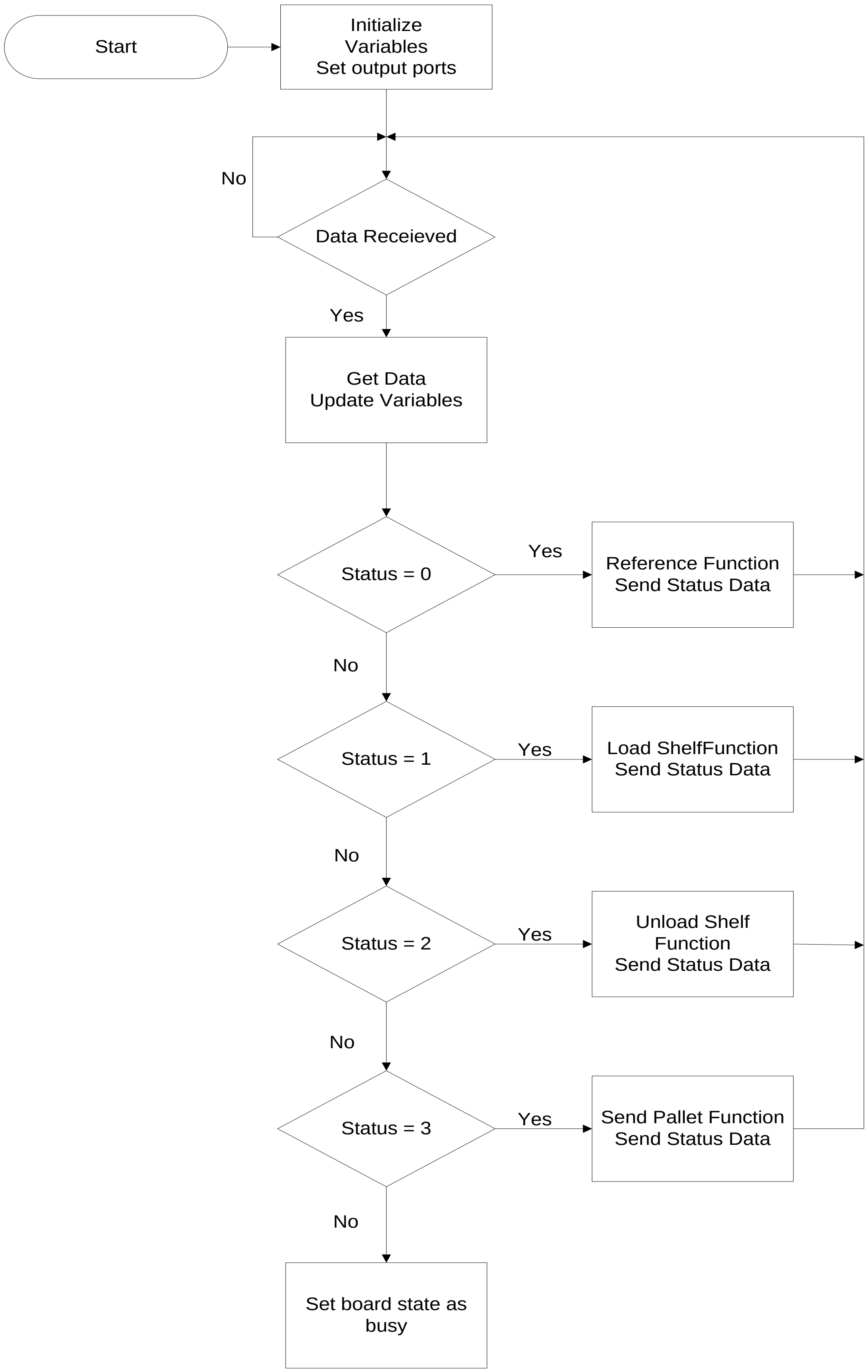


Figure 3. 12: Flowchart of the Stacker Board firmware

CHAPTER 4

CONTROL SOFTWARE

One of the objectives of this thesis is to develop a customized control program for the AS/RS housed in the Computer Integrated Manufacturing Laboratory (CIMLAB) located in Mechanical Engineering Department, Middle East Technical University. The control program is the intelligence part of the system and provides high level control. Also a graphical user interface (GUI) is developed in order to be used the system easily by end users. The control program is coded at .NET framework C# and it is designated as Miniaturized AS/RS Control.

During the design process of the Miniaturized AS/RS Control, applicability of control decisions to the current system is given special importance. These aforementioned control decisions are storage assignment, batching, dwell point and sequencing storage and retrieval requests. The batching cannot be applicable to the system since the stacker of the AS/RS is a single shuttle stacker. The Miniaturized AS/RS Control also keeps performance measurement information in order to compare experiment results.

Before giving details of the Miniaturized AS/RS Control, usage of the program should be explained briefly. Firstly, initial states of the shelves are set from GUI. After selection the control decisions (storage assignment, Dwell-Point location, sequencing storage and retrieval orders), arrival time and due dates of orders are set and then orders are inserted an order sequence. Then automation sequence is initiated. During an operation, performance measurement data and log can be tracked from GUI. If it is desired this information saved as a text file at the end of an operation.

4.1. Miniaturized AS/RS Control Methods and Events

The Miniaturized AS/RS Control contains mainly fields, methods and events. All the fields are set as public so users can access them but setting some of these fields are limited at the GUI. For example, arrival time of the orders cannot be set as an unrealistic value by end users. Types, modifiers and definitions of the fields are presented at Table 4.1. All the methods are classified with regard to usage and presented according to this classification. The methods and events are shown at Table 4.2 and Table 4.3.

Table 4. 1: Fields of the Miniaturized AS/RS Control

Type	Name	Definition
int[,]	Shelves	States of shelves
int[,]	CostOfEmptyShelfs	Distances of shelves to the I/O Point
int[,]	CostOfALoadedShelfs	Distances of Product A loaded shelves to the I/O Point
int[,]	CostOfBLoadedShelfs	Distances of Product B loaded shelves to the I/O Point
byte[]	Pic1Finish	Received data from the Handling Board
byte[]	Pic2Finish	Received data from the Stacker Board
byte[]	Pic1Send	Data to send to the Handling Board
byte[]	Pic2Send	Data to send to the Stacker Board
bool	Pic1Busy	State of the Handling Board
bool	Pic2Busy	State of the Stacker Board
int	calculatedX	Calculated column location of the stacker for loading product
int	calculatedY	Calculated row location of the stacker for loading a product
int	StackerPositionX	Current column location of the stacker
int	StackerPositionY	Current row location of the stacker
int	DwellPointX	Column location of dwell-point
int	DwellPointY	Row location of dwell-point
int	TotalCost	Overall distance of a shelf to the I/O Point
int	CurrentRequest	Sequence of current order
int[]	ClassBasedAX	Column locations of Product A loaded shelves
int[]	ClassBasedAY	Row locations of Product A loaded shelves
int[]	ClassBasedBX	Column locations of Product B loaded shelves
int[]	ClassBasedBY	Row locations of Product B loaded shelves

int	calculatedPalletX	Calculated column of the stacker for loading pallet to a shelf
int	calculatedPalletY	Calculated row of the stacker for loading pallet to a shelf
int[,]	CostOfPalletLoadedShelfs	Distances of pallet loaded shelves to the I/O Point
int[,]	CostOfPalletUnLoadedShelfs	Distances of empty shelves to the I/O Point
int	NumofTardyProduct	Number of tardy product
int	MaxNumofProductInQueue	Maximum number of product waited in the queue
datetime	ArrivalTime	Arrival time of orders
datetime	DueDate	Due dates of orders
datetime	HandlingTime	Entry time of orders to the system
datetime	FinishingTime	Exit time of orders from the system
timespan	TravelTime	Travel time of orders in the system
timespan	WaitingTime	Waiting time of orders in the queue

Table 4. 2: Methods of the Miniaturized AS/RS Control

Return Type	Name	Arguments
void	WriteText	void
void	ReadText	void
void	OpenSerialPorts	void
void	CloseSerialPorts	void
void	ArmReference	void
void	UnloadConveyor	void
void	LoadStacker	void
void	UnloadStacker	void
void	LoadConveyor	void
void	StackerReference	int 4, int 1
void	DwellPoint	int DwellPointX, int DwellPointY
void	PalletObtain	int calculatedPalletX, int calculatedPalletY
void	PalletGetRidOf	int calculatedPalletX, int calculatedPalletY
void	LoadShelf	int calculatedX, int calculatedY
void	UnloadShelf	int calculatedX, int calculatedY
void	ConveyerSend	void
void	CalculateForStore	void
void	CalculateForA	void
void	CalculateForB	void

int	CalculateCost	int StackerPositionX, int StackerPositionY, int x,int y
void	CalculateForUnloadPallet	void
void	CalculateForLoadPallet	void
void	StoreA	void
void	StoreB	void
void	RetrieveA	void
void	RetrieveB	void
void	UpdatePerformanceMeasurement	void
void	Automated	void
void	AcquireAutomationSequence	void

Table 4. 3: Events of the Miniaturized AS/RS Control

Return Type	Name	Arguments
void	SerialPort1_DataReceived	object sender, SerialDataReceivedEventArgs e
void	SerialPort2_DataReceived	object sender, SerialDataReceivedEventArgs e
void	btnReference_Click	object sender, EventArgs e
void	btnAutomated_Click	object sender, EventArgs e
void	btnPause_Click	object sender, EventArgs e
void	btnContinue_Click	object sender, EventArgs e
void	btnStoreA_Click	object sender, EventArgs e
void	btnRetrieveA_Click	object sender, EventArgs e
void	btnStoreB_Click	object sender, EventArgs e
void	btnRetrieveB_Click	object sender, EventArgs e
void	btnShelfState_Click	object sender, EventArgs e
void	btnSetOrderData_Click	object sender, EventArgs e
void	btnInsertOrder_Click	object sender, EventArgs e
void	btnRemove_Click	object sender, EventArgs e
void	btnControlSetting_Click	object sender, EventArgs e
void	btnSaveLog_Click	object sender, EventArgs e
void	btnClearLog_Click	object sender, EventArgs e
void	btnSavePerformance_Click	object sender, EventArgs e
void	btnClearPerformance_Click	object sender, EventArgs e

4.1.1. Serial Port Related Methods and Events

OpenSerialPorts: There are two boards in the system and they communicate with computer one by one. They use two different serial ports as COM1 and COM2. This method sets the properties of the serial port objects and opens them.

CloseSerialPorts: It closes the serial port objects.

SerialPort1_DataReceived: When the Handling Board finished its tasks, it sends one byte variable to the COM1 port. If the Miniaturized AS/RS Control receives a data from the COM1 port, this event occurs. In this event, the received data is assigned to the Pic1Finish and the Pic1Busy that indicates the Handling Board is ready or not is set as false.

SerialPort2_DataReceived: When the Stacker Board finished its tasks, it sends one byte variable to the COM2 port. If the Miniaturized AS/RS Control receives a data from the COM2 port, this event occurs. In this event, the received data is assigned to the Pic2Finish and the Pic2Busy that indicates the Stacker Board is ready or not is set as false.

4.1.2. Handling Board Related Methods and Events

ArmReference: It sets the value of the Pic1Send which is send to the Handling Board as 0 and this variable is send via COM1 port to the board. Before sending the Pic1Send, Pic1Busy is set as true. This variable indicates that a task is given to the board and it is operating this command. The Miniaturized AS/RS Control cannot give a task to the Handling Board until receiving a data from the board which informs the given task have finished. When the data is send to the Handling Board, the robot arm moves to the reference position.

UnloadPallet: Similar to the ArmReference method except this one the value of the Pic1Send is set as 1 and this variable is send to the Handling Board. When the data is send to the Handling Board, the robot arm moves to the input pallet, picks a product from input pallet and returns to the I/O Point respectively.

LoadStacker: Similar to the ArmReference method except this one the value of the Pic1Send is set as 2 and this variable is send to the Handling Board. When the data is send to the Handling Board, the robot arm moves to the stacker, loads a product to the stacker and returns to the I/O Point respectively.

UnloadStacker: Similar to the ArmReference method except this one the value of the Pic1Send is set as 3 and this variable is send to the Handling Board. When the data is send to the Handling Board, the robot arm moves to the stacker, unloads a product from the stacker and returns to the I/O Point respectively.

LoadPallet: Similar to the ArmReference method except this one the value of the Pic1Send is set as 4 and this variable is send to the Handling Board. When the data is send to the Handling Board, the robot arm moves to the output pallet, loads a part to the stacker and returns to the I/O Point respectively.

4.1.3. Stacker Board Related Methods and Events

StackerReference: This method takes two arguments as which column and row the stacker will go. Default value of the column and the row variables are 4 and 1 for this method since the reference position for the stacker is first row and fourth column of the rack. Before sending the Pic2Send to the Stacker Board, it is coded. Firstly, the value of the Pic2Send is set as 0. Then column variable multiplied by 4 and row variable multiplied by 16 and these variables are added to the Pic2Send. At the board side, the received data is decoded. Also, this method updates StackerPositionX and StackerPositionY which hold current stacker location. Eventually, Pic2Busy that indicates a task is given to the Stacker Board and it is operating this command is set as true and then Pic2Send is send via COM2 port to the board. The Miniaturized AS/RS Control cannot give a task to the Stacker Board until receiving a data from the board which informs the given task have finished. When the Pic2Send is send to the Stacker Board, the stacker moves to the reference position.

DwellPoint: Similar to the StackerReference method except this method takes different column and row value. When the Pic2Send is send to the Stacker Board, the stacker moves to the Dwell-Point which is determined from GUI by end users.

LoadShelf: Similar to the StackerReference method except this method sets initial value of the Pic2Send as 1 and then codes the variable. When the data is send to the Stacker Board, the stacker moves to the calculated column and row location and load a product to the calculated shelf respectively.

UnloadShelf: Similar to the StackerReference method except this method sets initial value of the Pic2Send as 2 and then codes the variable. When the data is send to the Stacker Board, the stacker moves to the calculated column and row location and unload a product from the shelf respectively.

OutputPalletSend: It sets the value of the Pic2Send as 3. When the data is send to the Handling Board, the loaded output pallet moves to the operator.

4.1.4. Selection of Shelves for Storage and Retrieval Orders Related Methods and Events

CalculateCost: This method is used for Closest Open Location Storage Assignment and calculates minimum travel time of the stacker between the current stacker location and possible next stacker location. It takes four parameters as the current row location, the current column location, possible next row location and possible next column location of the stacker respectively. The value of the travel time at the vertical direction is multiplied by 3 since speed of the stacker at the horizontal direction is three times faster than the speed of the stacker at the vertical direction. Then minimum of these values is assigned to the cost of the possible next stacker location.

CalculateForStore: Selects the storage location for a storage order in two ways according to the selected storage location assignment. If Closest Open Location is selected at the GUI, the total travel time of the stacker for each empty shelves are calculated by calling the CalculateCost method and then the storage location of the current order is selected in such a way that the travel time of the stacker will be minimum. For storage operation, the stacker moves to the selected shelf from the current location of stacker in order to picking the pallet and then moves to the I/O point. After the stacker loaded by the robot arm, it moves to the selected shelf for loading the product to the shelf and this location becomes the current location of the stacker. The total travel time for each empty shelf is calculated as travel time between current stacker location and a possible storage location and then added two times of the travel time between the possible storage location and the I/O point. If Class Based Storage Assignment is selected at the GUI, shelves are divided in two classes for two different product types. Storage location is selected randomly in empty shelves of reserved class.

CalculateForRetrieveA: Similar to the CalculateForStore method except this method selects the retrieval location of an order for the A type product.

CalculateForRetrieveB: Similar to the CalculateForStore method except this method selects the retrieval location of an order for the B type product.

CalculateForUnloadPallet: If Class Based Storage Assignment is selected at the GUI, this method selects the pallet location for a storage order. The fourth column of the rack is devoted to the pallets and it selects the pallet which is closest to the I/O Point. If there is not a pallet on the shelves of the fourth column, it selects a pallet which is closest to the I/O Point from pallet loaded shelves.

CalculateForLoadPallet: Similar to the CalculateForUnloadPallet method except this method selects the pallet location for a retrieval order.

4.1.5. Input Data of Orders Related Methods and Events

btnShelfState_Click: After entering all of states of the shelves to the related textboxes, this event writes the shelves states to a text file which name is "ShelfState.txt". In the text file, first seven lines corresponds to first row of the rack, second seven lines corresponds to second row of the rack and last seven lines corresponds to third row of the rack. In this event, this text file is read and states of shelves are assigned to the Shelf array which keeps the states of the shelves throughout a runtime of the program.

btnControlSetting_Click: This event sets the control decisions which are selected at the GUI. This control decisions include storage assignment, Dwell-Point location and sequencing order.

btnInsert_Click: Order types, arrival time of the orders and due date of the retrieval orders are identified. After all of the parameters of an order are entered, this event inserts the order to the order sequence.

btnRemove_Click: If an order is entered incorrectly, this method removes the incorrect order from the order sequence.

btnSetOrderData_Click: This event parses the inserted order data and assigns these data to the member of Order class which includes OrderType, ArrivalTime and DueDate. If sequenced order is selected at the GUI, orders are sequenced as defined by user. This event sequences the order according to the arrival time of the orders for the First In First Out order sequencing rule. Then orders of the sequenced order are added to an object of the queue class which name is AutomationSequence. This queue serves as First In First Out and processed orders are exited from the queue.

4.1.6. Automation Related Methods and Events

StoreA: When the current order is a storage order for the A type product this method is called. This method calls the Handling Board and the Stacker Board related methods in order to complete a storage operation. Also, it calculates some of the performance data as waiting time in the queue and travel time of the order. In the method, the each part of the physical system is logged with time of occurrences on the Log textbox.

StoreB: Similar to the StoreA method except this method is called for storage order of the B type product.

RetrieveA: Similar to the StoreA method except this method is called for retrieval order of the A type product.

RetrieveB: Similar to the StoreA method except this method is called for retrieval order of the B type product.

Automated: This method controls current member of AutomationSequence and decided the order type. It waits until the arrival time of the current order and then calls related method according to the order type. In the meantime, the current order is deleted from the queue.

btnAutomated_Click: Starts a thread for Automated method since it runs for a long time. If it runs in the same thread with the windows form of the Miniaturized AS/RS Control, user cannot access the GUI.

btnReference_Click: Calls ArmReference and StackerReference method in order to referencing the parts of the physical system.

btnPause_Click: This event keeps the Miniaturized AS/RS Control waiting until the btnContinue_Click event occurs. When btnPause_Click event occurred, the program can wait after completion of the current order.

btnContinue_Click: If this event occurs after the btnPause_Click event occurred, the program resumes from the remaining order queue.

4.1.7. Performance and Log Related Methods and Events

UpdatePerformanceMeasurement: Travel time of each order, waiting time in the queue of each order, average travel time, average interarrival time, number of product processed per minute, number of product waiting in the queue, number of tardy products, time weighted average queue size and the maximum number of product in the queue are selected as performance measures for studying the performance of the physical system. These parameters are updated and shown at the GUI after finishing each order during runtime of the Miniaturized AS/RS Control.

Travel time and waiting time in the queue of each order are calculated in the Store and Retrieve methods and the others are calculated in this method. Also this method updates the state of the shelves at the GUI during an operation.

btnPerformanceSave: Saves all of the performance measurement information to a text file which name is Performance Measurement.

btnPerformanceClear: Clears all the textboxes and rich textboxes related performance measurement at the GUI.

btnLogSave: Saves all the data that were logged on the Log rich textbox to a text file which name is Operation Log.

btnLogClear: Clears the Log rich textbox at the GUI.

4.2. Graphical User Interface(GUI)

During the design stages of the Miniaturized AS/RS Control several GUIs are designed and one of them is presented in this subsection (Figure 4.1). The GUI is coded at C# as a Windows form application.

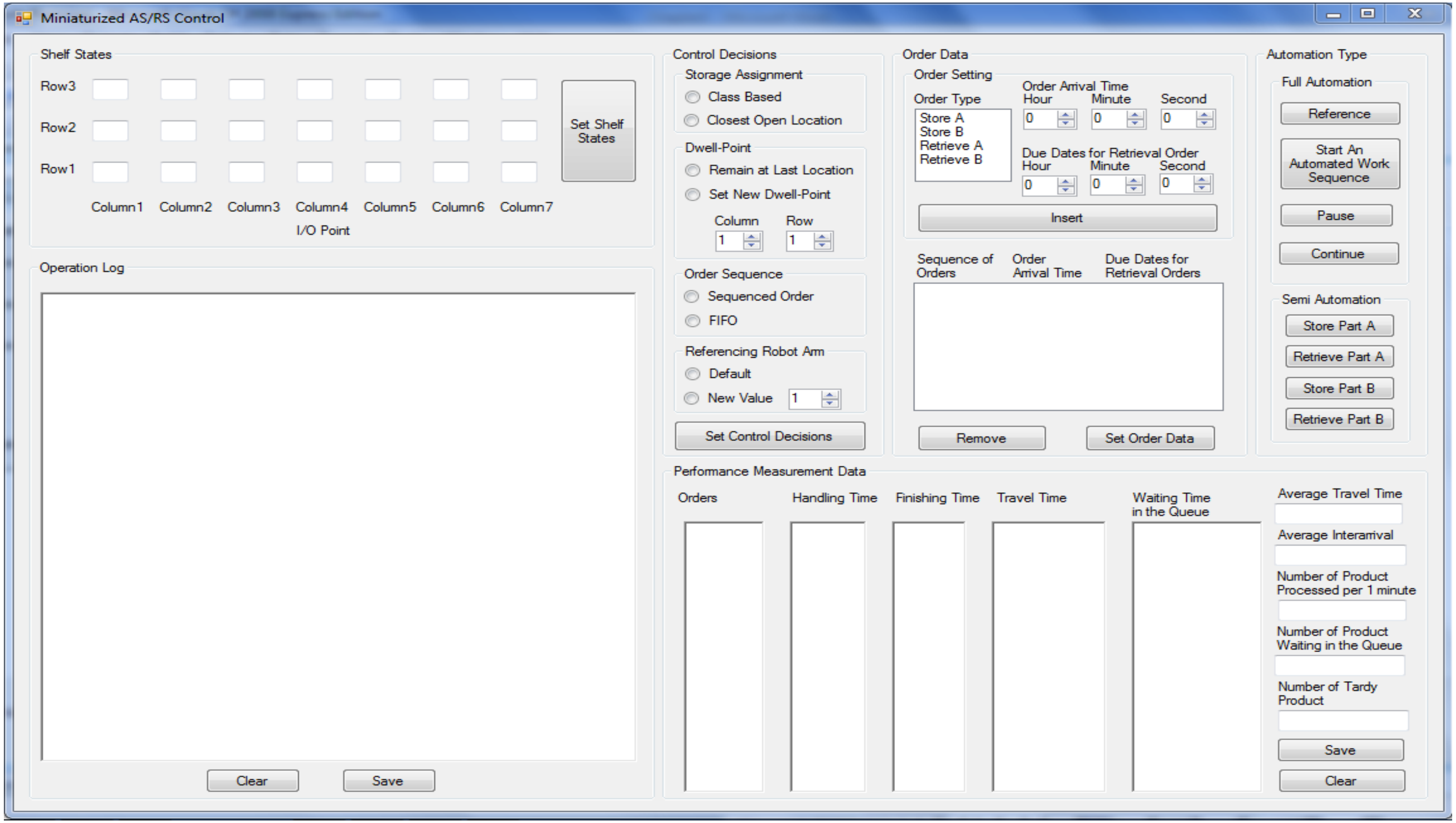


Figure 4. 1: Sample GUI

For identifying the initial state of the shelves there is a group control (Figure 4.2). Letters A, B, E, P are entered to the related textboxes for A type product loaded shelves, B type product loaded shelves, empty shelves and pallet loaded shelves, respectively. There is a button in this group control for setting the states. Also current state of the shelves is observed from this part during an operation.

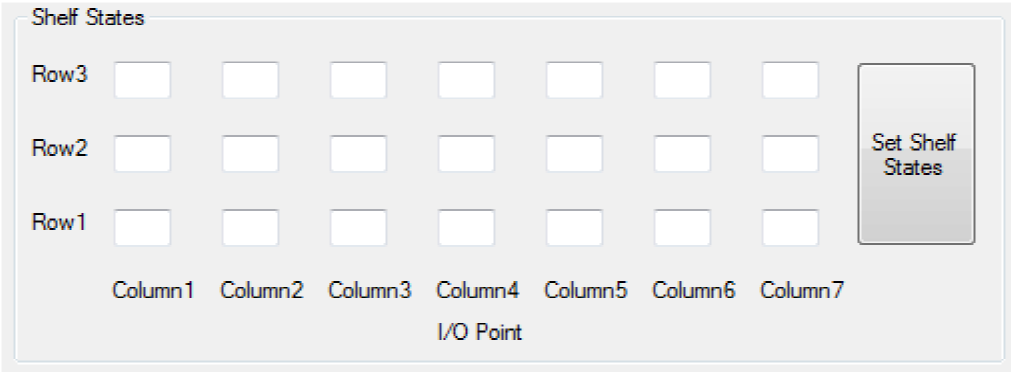


Figure 4. 2: States of shelves related controls of the GUI

There is a group control for setting the four control decisions as storage assignment rule, Dwell-Point location, Order sequence rule and referencing the robot arm (Figure 4.3). Except referencing the robot arm, control decisions about AS/RSs are mentioned before so they are not explained again. The robot arm position is controlled by three encoders at three axes as open loop control. During a long working process of the physical system, error is accumulated and the robot arm may be located incorrect position. In order to avoiding this error, the robot arm must be referenced in a certain order period. Default value of this period is five orders. If it is desired to set a new value for this period, there is a numeric up down control. When defined all the control decisions, selections are set by Set Control Decisions button.

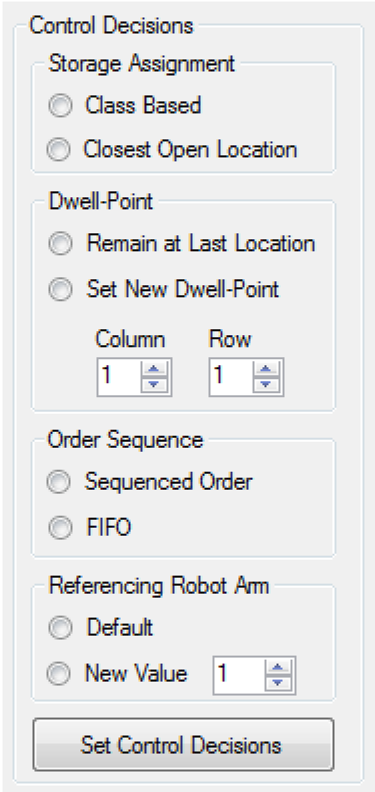


Figure 4. 3: Control decisions related controls of the GUI

Order type, order arrival time and due date of retrieval orders are set and inserted in an order queue in the order data group control. If an order is set incorrectly, there is a remove button for removing this order from the order queue.

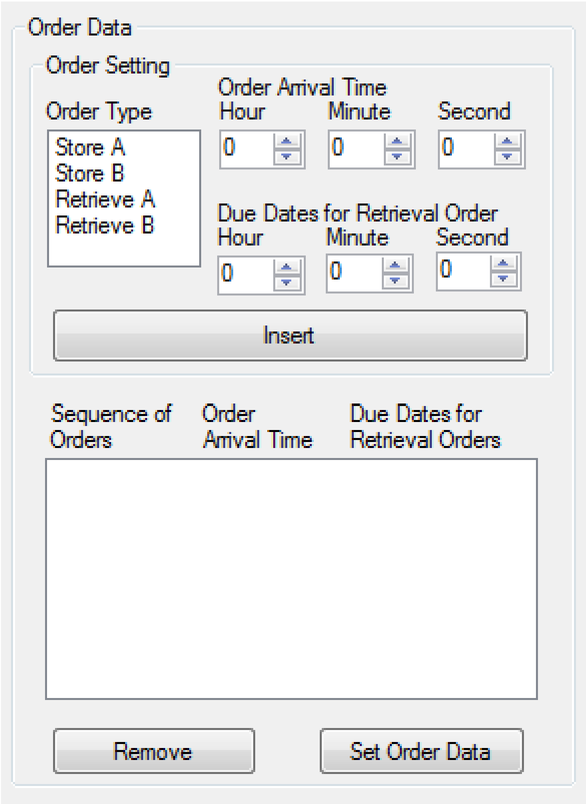


Figure 4. 4: Order data related controls of the GUI

There are two automation types as full automation and semi automation. In the full automation, Miniaturized AS/RS Control processed each order which is in the order queue one by one. Also there is a pause button and a continue button for controlling the full automation process. In the semi automation, a user selects the order type and controls an operation manually by buttons which is devoted to each order type. For either type of automation, physical system must be referenced by reference button.

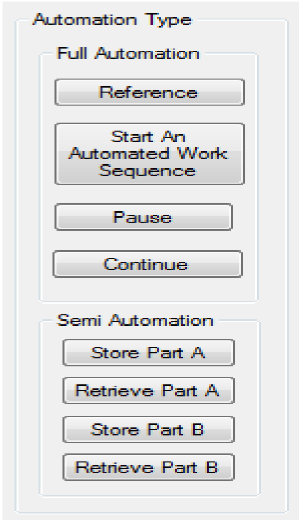


Figure 4. 5: Automation type related controls of the GUI

There is a group control for monitoring performance measurement data during an operation, (Figure 4.6). After finishing an order these data are updated and presented in this group control. If it is desired, performance data can be saved to a text file by Save button at the end of an operation.

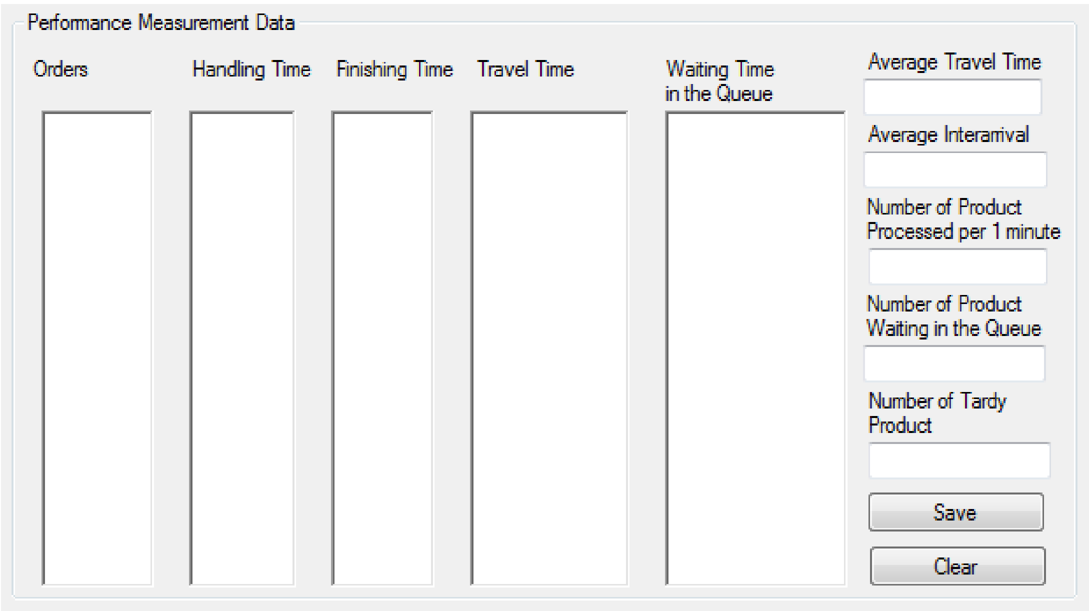


Figure 4. 6: Performance Measurement Data related controls of the GUI

CHAPTER 5

CASE STUDY

Two experiments are run for demonstrating capacity and versatility of the control system. In these experiments 10 orders are processed and orders data presented in Table 5.1. Order types, interarrival time, time interval between arrival time and due dates for the retrieval orders and initial state of the shelves are the same for these experiments but different control decisions are used.

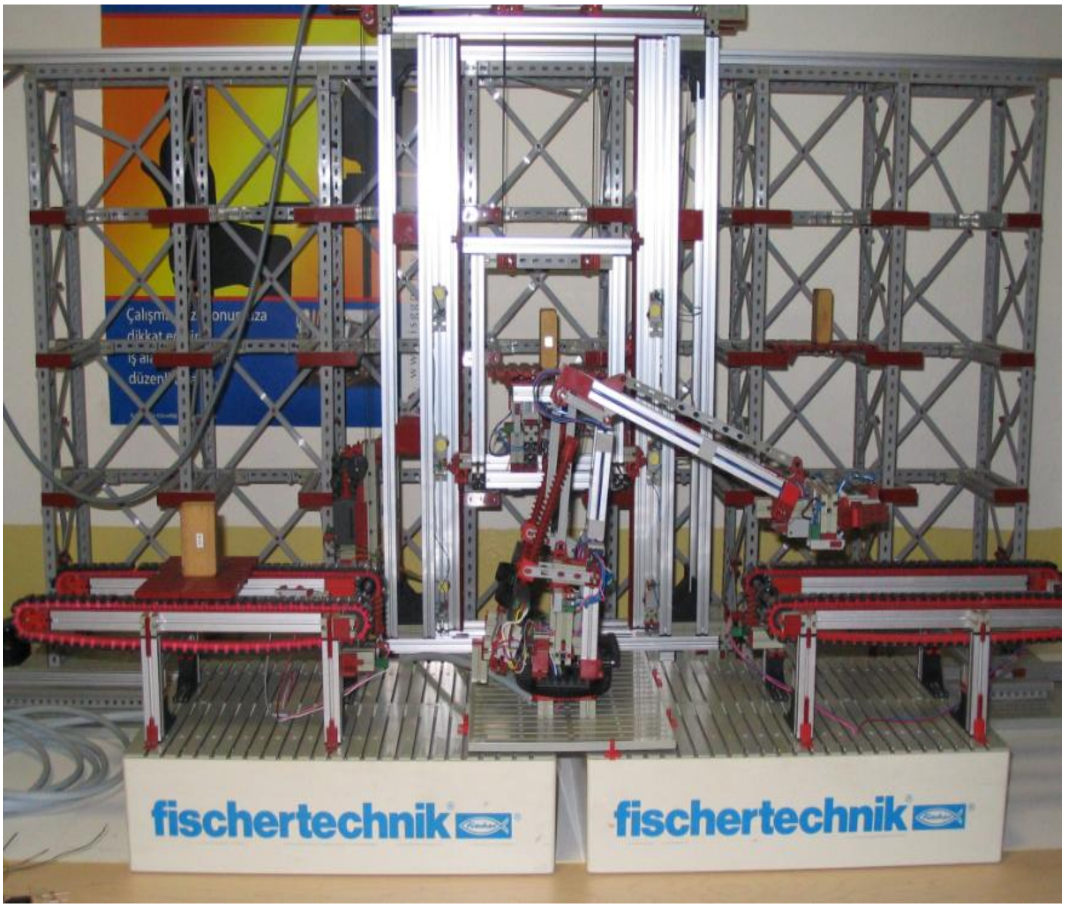


Figure 5. 1: Experiment setup

Table 5. 1: Orders Data for the experiments

Order Type	Product Type	Interarrival Time (minute)	Time interval between arrival time and due date (minute)
Retrieve	A	0	5
Store	B	0	-
Retrieve	A	1	5
Retrieve	A	5	5
Store	A	0	-
Retrieve	B	1	3
Store	B	0	-
Retrieve	B	0	3
Retrieve	A	4	4
Store	A	1	-

Table 5. 2: Initial state of the shelves for the experiments

Column 3	Empty	Product A	Product A	Pallet	Product B	Empty	Empty
Column 2	Empty	Product A	Empty	Pallet	Empty	Empty	Product B
Column 1	Product A	Empty	Product A	Pallet	Empty	Product B	Empty
	Row 1	**Row 2**	**Row 3**	**Row 4**	**Row 5**	**Row 6**	**Row 7**

5.1. Experiment 1

For implementing the experiment necessary hardware connection is should be made. Connection details and lab sheet are presented at Appendix B. After the connections are made the Miniaturized AS/RS Control is run and order data, initial state of the shelves and control decisions are entered from GUI. For this experiment, Class-Based storage assignment which divides rack in two classes for A type product and B type product is selected. Products are stored to the own class and retrieved from own class randomly. Sequenced order is selected as an order sequence rule. In this type of order sequence the program processed the orders regardless of the arrival time of orders. The fourth row and second column is set as the Dwell-Point. These control decisions, orders data and initial state of the shelves are to be set at the GUI before starting the operation (Figure 5.2). Finally, the physical system is referenced by reference button and the automated order sequence is started.

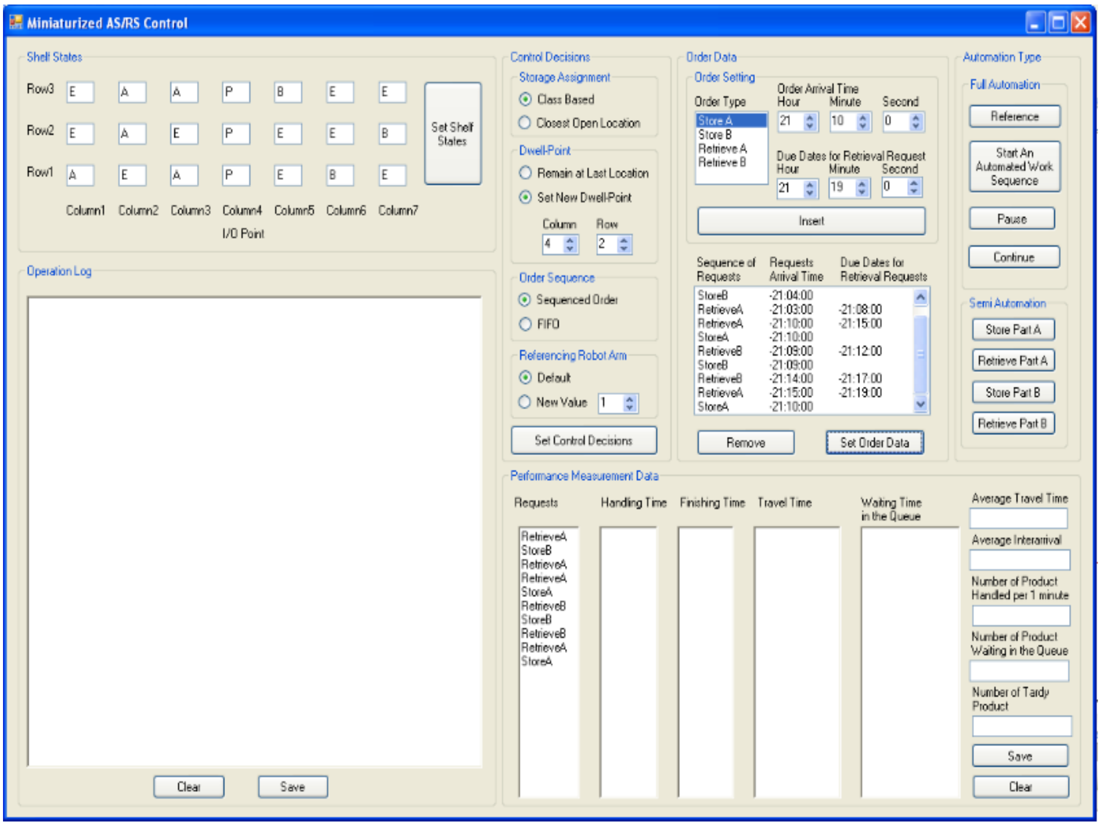


Figure 5. 2: Experiment GUI for the Experiment 1 before starting the operation

When operation is started performance data are calculated and written at the GUI after completing every order. Also operation log are written in order to tracking parts of the physical system situations. Remaining orders are followed from GUI. Experiment GUI at the runtime of the program is presented in Figure 5.3.

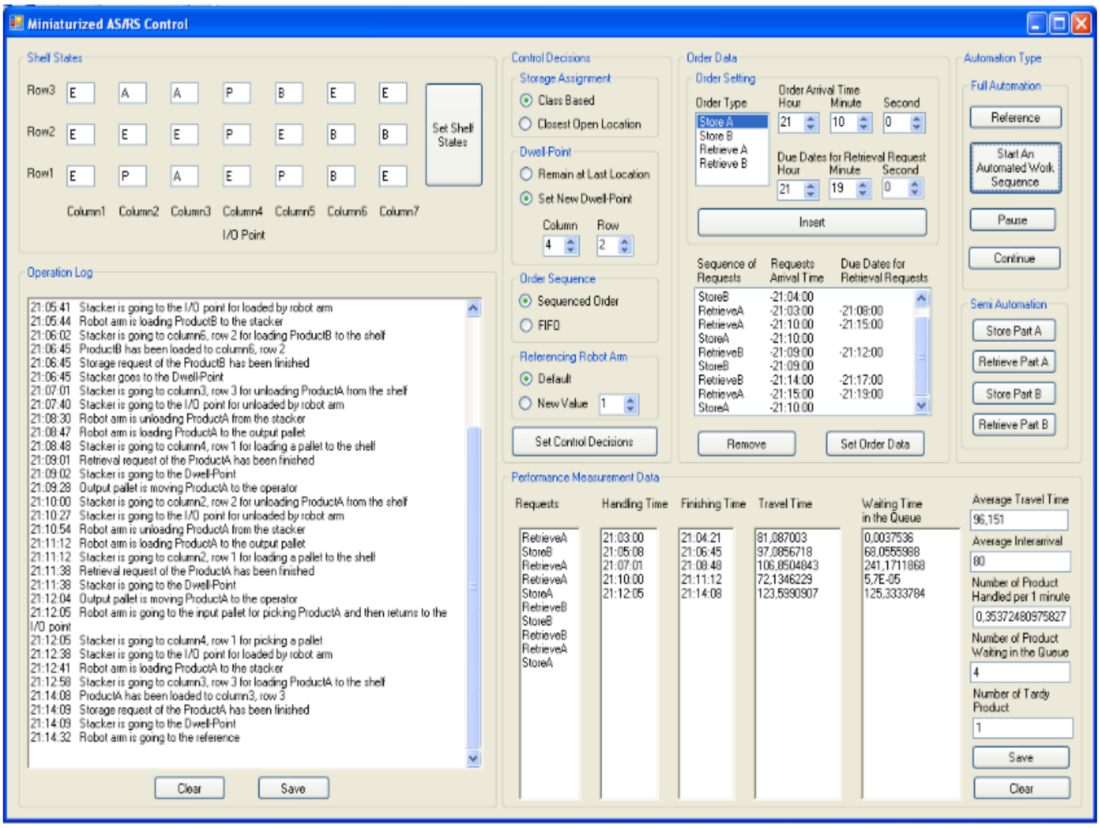


Figure 5. 3: Experiment GUI for the Experiment 1 at the runtime of the program

On completing the experiment, all the experiment information is displayed at the GUI. If it is desired performance measurement data and the operation log are saved to text files.

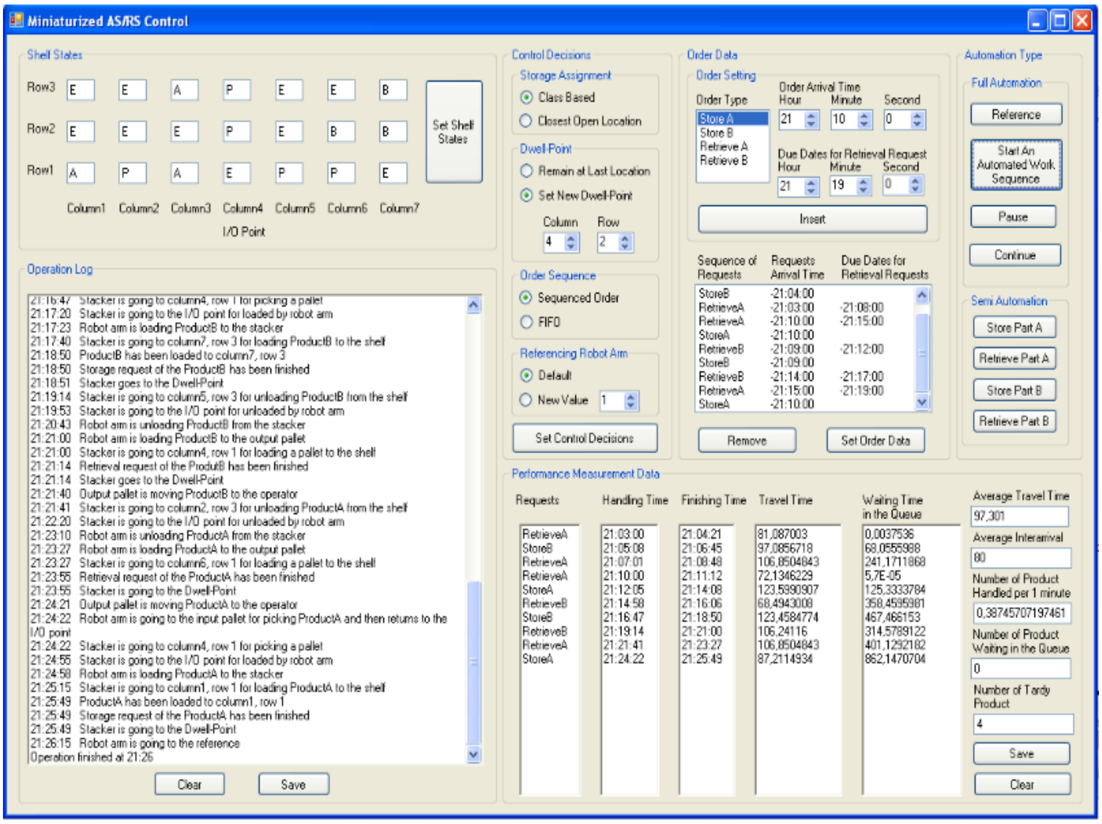
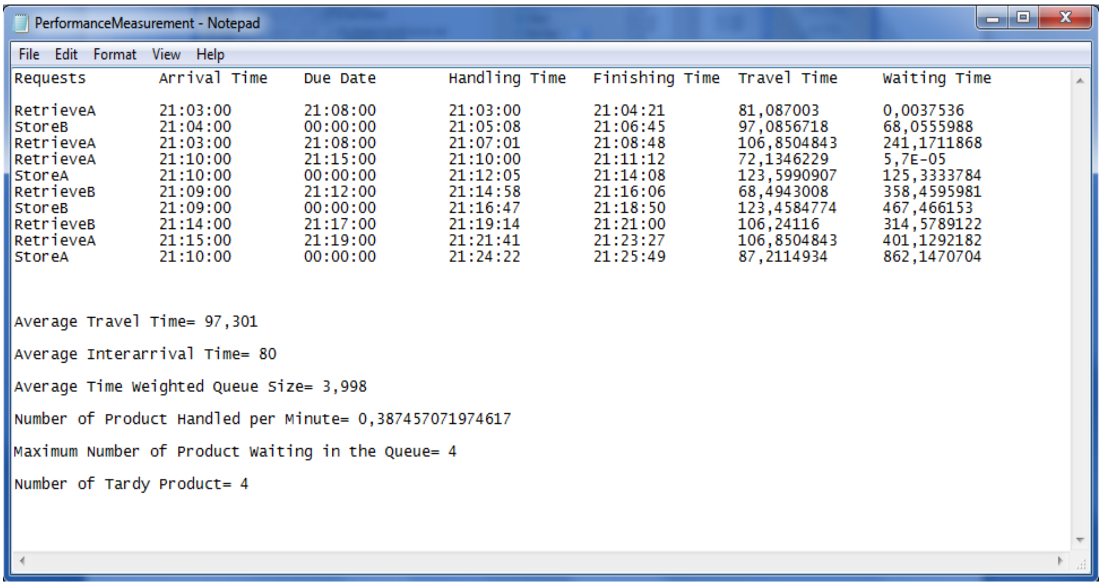


Figure 5. 4: Experiment GUI for the Experiment 1 at the end of the operation



PerformanceMeasurement - Notepad

Requests	Arrival Time	Due Date	Handling Time	Finishing Time	Travel Time	Waiting Time
RetrieveA	21:03:00	21:08:00	21:03:00	21:04:21	81,087003	0,0037536
StoreB	21:04:00	00:00:00	21:05:08	21:06:45	97,0856718	68,0555988
RetrieveA	21:03:00	21:08:00	21:07:01	21:08:48	106,8504843	241,1711868
RetrieveA	21:10:00	21:15:00	21:10:00	21:11:12	72,1346229	5,7E-05
StoreA	21:10:00	00:00:00	21:12:05	21:14:08	123,5990907	125,3333784
RetrieveB	21:09:00	21:12:00	21:14:58	21:16:06	68,4943008	358,4595981
StoreB	21:09:00	00:00:00	21:16:47	21:18:50	123,4584774	467,466153
RetrieveB	21:14:00	21:17:00	21:19:14	21:21:00	106,24116	314,5789122
RetrieveA	21:15:00	21:19:00	21:21:41	21:23:27	106,8504843	401,1292182
StoreA	21:10:00	00:00:00	21:24:22	21:25:49	87,2114934	862,1470704

Average Travel Time= 97,301

Average Interarrival Time= 80

Average Time Weighted Queue Size= 3,998

Number of Product Handled per Minute= 0,387457071974617

Maximum Number of Product Waiting in the Queue= 4

Number of Tardy Product= 4

Figure 5. 5: Results of the Experiment 1

When the performance parameters were calculated manually for verification purposes, it is seen that results of the experiment are correct. There was no unexpected situation related to the control program and the physical system during the experiment.

5.2. Experiment 2

For this experiment orders data, initial state of the shelves and experiment procedure are the same as in Experiment 1. But different control decisions are selected for the experiment. Closest Open Location is selected as a storage assignment rule in this experiment. In this policy, products can be stored in any shelf of the rack regardless of the product type. The shelf location is selected for storage order as minimizing the travel time of the stacker between the I/O Point and the selected shelf. For retrieval orders, shelf locations are selected by the same logic. In this experiment, the Dwell-Point for the stacker was not set so after finishing every order, the stacker remains at its last location. As for the sequencing rule, First In First Out rule is selected. This rule sequenced orders according to the arrival time of the orders and orders are processed in this sequence. These control decisions, orders data and initial state of the shelves are to be set at the GUI before starting the operation (Figure 5.6). Finally, the physical system is referenced by reference button and the automated order sequence is started.

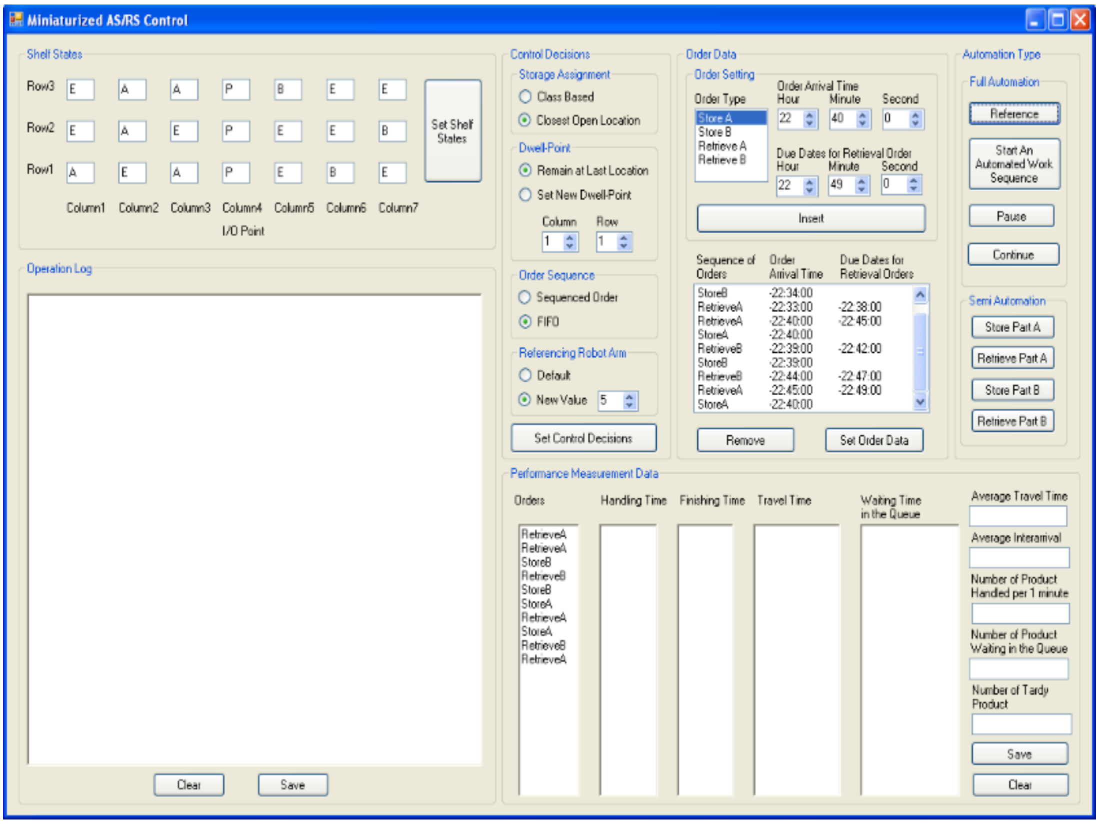


Figure 5. 6: Experiment GUI for the Experiment 2 before starting the operation

Experiment GUI for this experiment at the end of the operation and experiment results are presented in Figure 5.7 and Figure 5.8, respectively.

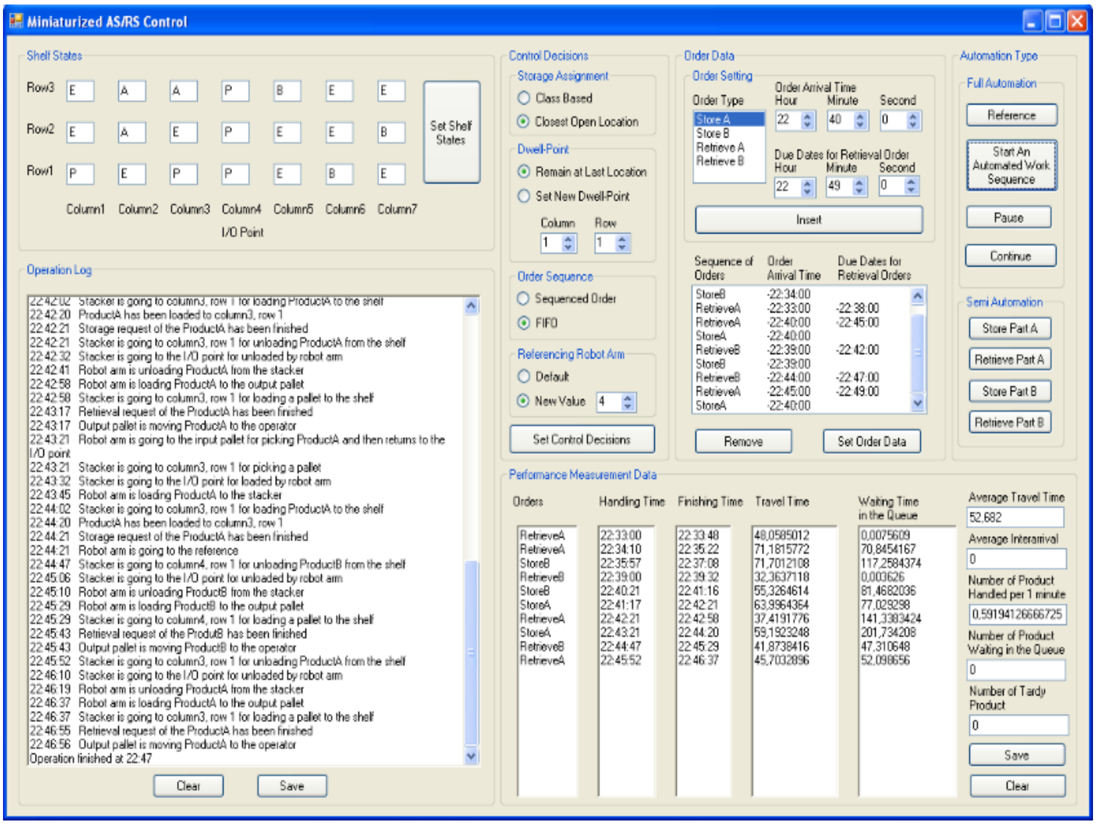
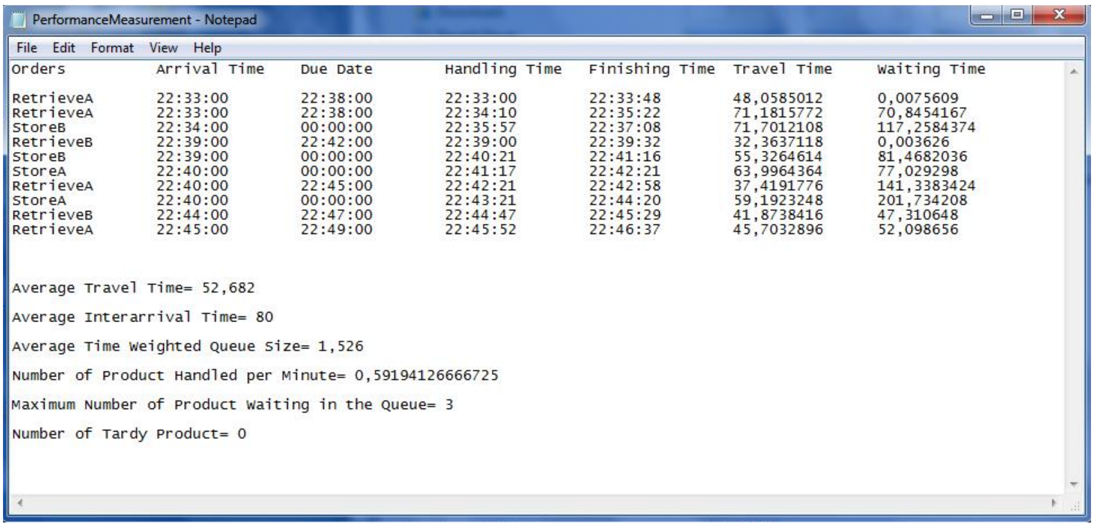


Figure 5. 7: Experiment GUI for the Experiment 2 at the end of the operation


PerformanceMeasurement - Notepad

Orders	Arrival Time	Due Date	Handling Time	Finishing Time	Travel Time	Waiting Time
RetrieveA	22:33:00	22:38:00	22:33:00	22:33:48	48,0585012	0,0075609
RetrieveA	22:33:00	22:38:00	22:34:10	22:35:22	71,1815772	70,8454167
StoreB	22:34:00	00:00:00	22:35:57	22:37:08	71,7012108	117,2584374
RetrieveB	22:39:00	22:42:00	22:39:00	22:39:32	32,3637118	0,003626
StoreB	22:39:00	00:00:00	22:40:21	22:41:16	55,3264614	81,4682036
StoreA	22:40:00	00:00:00	22:41:17	22:42:21	63,9964364	77,029298
RetrieveA	22:40:00	22:45:00	22:42:21	22:42:58	37,4191776	141,3383424
StoreA	22:40:00	00:00:00	22:43:21	22:44:20	59,1923248	201,734208
RetrieveB	22:44:00	22:47:00	22:44:47	22:45:29	41,8738416	47,310648
RetrieveA	22:45:00	22:49:00	22:45:52	22:46:37	45,7032896	52,098656

Average Travel Time= 52,682

Average Interarrival Time= 80

Average Time Weighted Queue Size= 1,526

Number of Product Handled per Minute= 0,59194126666725

Maximum Number of Product Waiting in the Queue= 3

Number of Tardy Product= 0

Figure 5. 8: Results of the Experiment 2

When the performance parameters were calculated manually for verification purposes, it is seen that results of the experiment are correct. There was no unexpected situation related to the control program and the physical system during the experiment.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

This chapter presents outcomes of the study and discusses some possible future works for extending this study.

A survey on AS/RS related laboratory facilities and AS/RS related literature survey is made in order to defining the design criteria for the control system of the current AS/RS. As a result of these surveys, applicability of the control decisions are adopted as first design criteria. Also, the control system should be flexible and versatile in terms of adding new equipments to the system or configuring locations of the system parts. Considering these design criteria, two boards and their firmware are developed as a low level control system. Besides, a high level control system is developed using C#. This high level control system implements some of AS/RS related control decisions and calculates performance measurement data of experiments. A GUI is designed in order to using the system easily and analyzing the system in real time.

After the developed and manufactured high level and low level control system, overall system is tested. Two experiments are implemented and observed that an unexpected situation related control system and the physical system is not occurred during the experiments.

The main improvements and future research topics related this study are given below:

- In the current system the robot arm and the output pallet are controlled by different boards. In some cases, this causes to waiting of the output pallet. So the robot arm, the input and output pallets should be controlled by the same board.

- In the developed system, the boards are communicates the high level control system one at a time. In terms of the expandability of the system, peripheral communication protocols as SPI or I2C should be used.
- Product types are introduced from GUI in the current system. RFID technologies or image processing should be used for product recognition.
- The current software is developed for two product types. Product types should be generalized in order to run various experiments.
- In the developed software some of the AS/RS related control decisions are used. In the literature, there are a lot of static and dynamic control algorithms and these algorithms should be used in order to implement various experiments.
- Shuffling algorithms which change locations of empty pallets should be developed for the control software.
- The current system is very slow (3-4 cm/sec) and the DC motors of the system should be changed by more powerful motors.

REFERENCES

Bozer, Y.A., White, J.A., (1984), “Travel-time models for automated storage/retrieval systems”, IIE Transactions 16 (4), pp. 329–338.

BUFAIM, Home page for Flexible Automation and Intelligent Manufacturing Systems Laboratory, Boğaziçi University, http://www.bufaim.boun.edu.tr/Default.aspx, last visited in January 2012.

Chang, D.T., Wen, U.P., (1997), “The impact on rack configuration on the speed profile of the storage and retrieval machine”, IIE Transactions 29, pp. 525–531.

De Koster, R., van der Poort, E., Wolters, M., (1999), “Efficient order batching methods in warehouses”, International Journal of Production Research 37 (7), pp. 1479–1504.

Eben-Chaime, M., Pliskin, N., (1997), “Operations management of multiple machine automatic warehousing systems”, International Journal of Production Economics 51, pp. 83–98.

Egbelu, P.J.,(1991), “Framework for dynamic positioning of storage/retrieval machines in an automated storage and retrieval system”, ”, International Journal of Production Research 29 (1), pp. 17–37.

Egbelu, P.J., Wu, C.T., (1993), “A comparison of dwell point rules in an automated storage/retrieval system”, International Journal of Production Research 31 (11), pp. 2515–2530.

Elsayed, E.A., (1981), “Algorithms for optimal material handling in automatic warehousing systems”, International Journal of Production Research 19 (5), pp. 525–535.

Elsayed, E.A., Stern, R.G., (1983), “Computerized algorithms for order processing in automated warehousing systems”, International Journal of Production Research 21 (4), pp. 579–586.

Farrukh, M.A, Gardezi, A.R., Hayat, N., (2009) “Design, Development and Analysis of Automated Storage and Retrieval System with Single and Dual Command Dispatching using MATLAB”, World Academy of Science, Engineering and Technology 60, pp. 91-95.

Goetschalckx, M., Ratliff, H.D., (1990), “Shared storage policies based on the duration of stay of unit loads”, Management Science 36, pp. 1120–1132.

Graves, S.C., Hausman, W.H., Schwarz, L.B., (1977). “Storage-retrieval interleaving in automatic warehousing systems”, Management Science 23 (9), 935–945.

Han, M.H., McGinnis, L.F., Shieh, J.S., White, J.A., (1987), “On sequencing retrievals in an automated storage/retrieval system”, IIE Transactions 19 (1), pp. 56–66.

Hausman, W.H., Schwarz, L.B., Graves, S.C., (1976), “Optimal storage assignment in automatic warehousing systems”, Management Science 22 (6), pp. 629–638.

Heskett, J.L., (1963), “Cube-per-order index – a key to warehouse stock location”, Transportation and Distribution Management 3, pp. 27–31.

Heskett, J.L., (1964), “Putting the cube-per-order index to work in warehouse layout”, Transportation and Distribution Management 4, pp. 23–30.

Hwang, H., Lee, M.K., (1988), “Order batching algorithms for a man-on-board automated storage and retrieval system”, Engineering Costs and Production Economics 13, pp. 285–294.

Intelitek, Educational CIM Setup, http://www.intelitek.com/Products.asp?CategoryID=135&Industrial=&Education=yes&category_str_id=5;135, last visited in January 2012.

Kayalıgil S., Altınçay H., (1996) “Küçültülmüş Ölçekli Bir Otomatik İmalat Hücre Modelinde Parça Yükleme Sıralaması Deneyi” Endüstri Mühendisliği, Cilt.7, pp. 3–9. (in Turkish)

Kulturel, S., Ozdemirel, N.E., Sepil, C., Bozkurt, Z., (1999), “Experimental investigation of shared storage assignment policies in automated storage/ retrieval systems.”, IIE Transactions 31, pp. 739–749.

Lee, Y.H., Lee, M.H., Hur, S., (2005), “Optimal design of rack structure with modular cell in AS/RS”, International Journal of Production Economics 98, pp. 172–178.

Material Handling Industry of America, Material Handling Equipment, http://www.ise.ncsu.edu/kay/mhetax/StorEq/index.htm, last visited in January 2012.

O'Shea, L., (2007), "Development of an Automated Storage and Retrieval System in a Dynamic Knowledge Environment", Master of Science Thesis, Waterford Institute of Technology, pp. 37-75.

Pan, C.H., Liu, S.Y., (1995), "A Comparative Study of Order Batching Algorithms", Omega, Int. J. Mgmt Sci. 23, (6), pp. 691-700.

Potrc, I., Lehrer, T., Kramberger, J., Sraml, M., (2004), "Simulation model of multi-shuttle automated storage and retrieval systems", Journal of Materials Processing Technology 157–158, pp. 236–244.

Roodbergen, K.J., Vis, I. F. A., (1998), "A survey of literature on automated storage and retrieval systems", European Journal of Operational Research 194, pp. 343-362.

Schwarz, L.B., Graves, S.C., Hausman, W.H., (1978), "Scheduling policies or automatic warehousing systems: Simulation results", AIIE Transactions 10 (3), pp. 260–270

Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco, J.M.A., (2003), Facilities Planning, third Ed. John Wiley & Sons.

Warehouse Modernization and Layout Planning Guide, (1985) Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, pp. 8–17.

Wen, U.P., Chang, D., Chen, S.P., (2001), "The impact of acceleration/deceleration on travel-time models in class-based automated S/R systems", IIE Transactions 33, pp. 99–608.

APPENDIX A

COMPONENTS OF THE AS/RS

Individual hardware components of the AS/RS in Computer Integrated Manufacturing Laboratory (CIMLAB) were explained in Section 3.1.1. The photographs of the system are presented for understanding the main characteristics of the system better in this appendix.

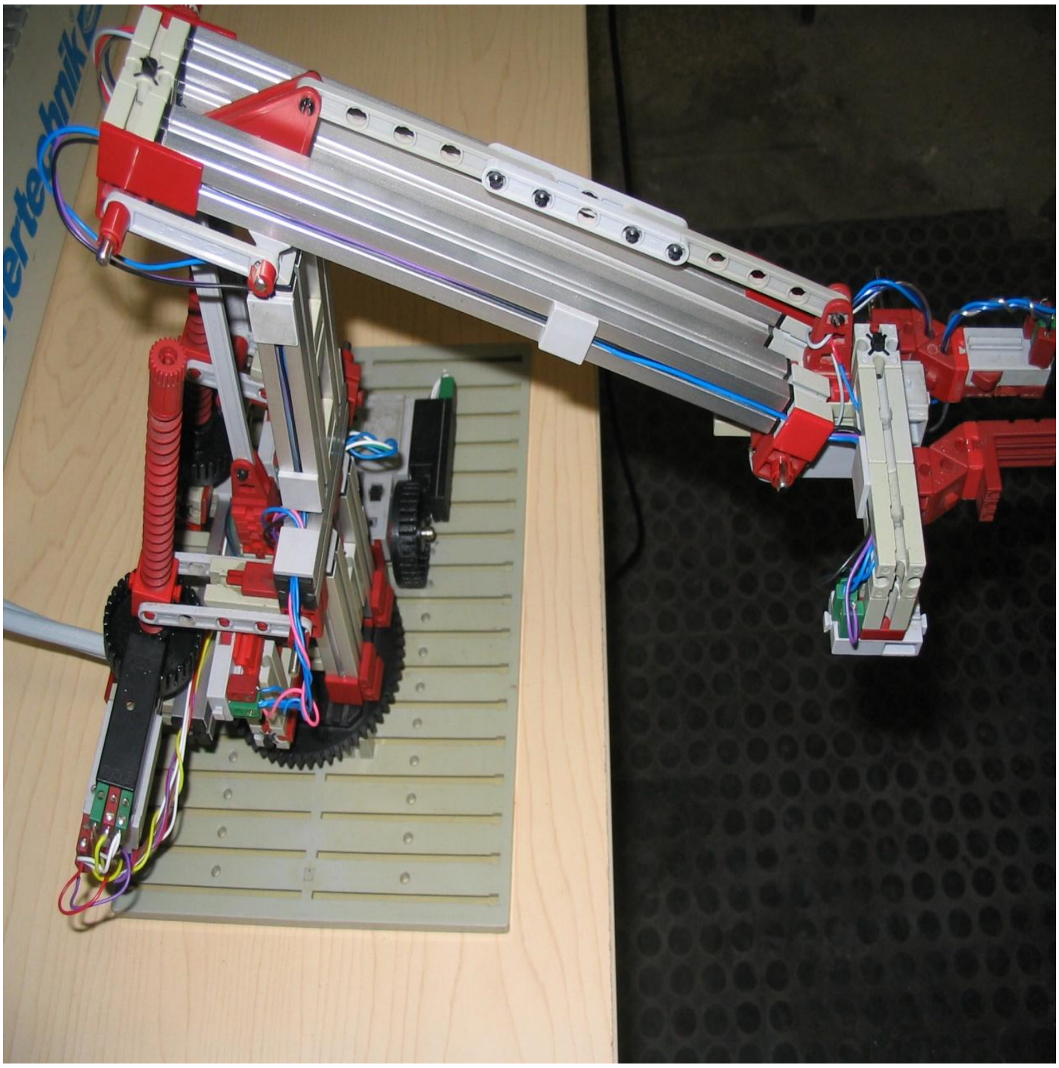


Figure A. 1: The robot arm

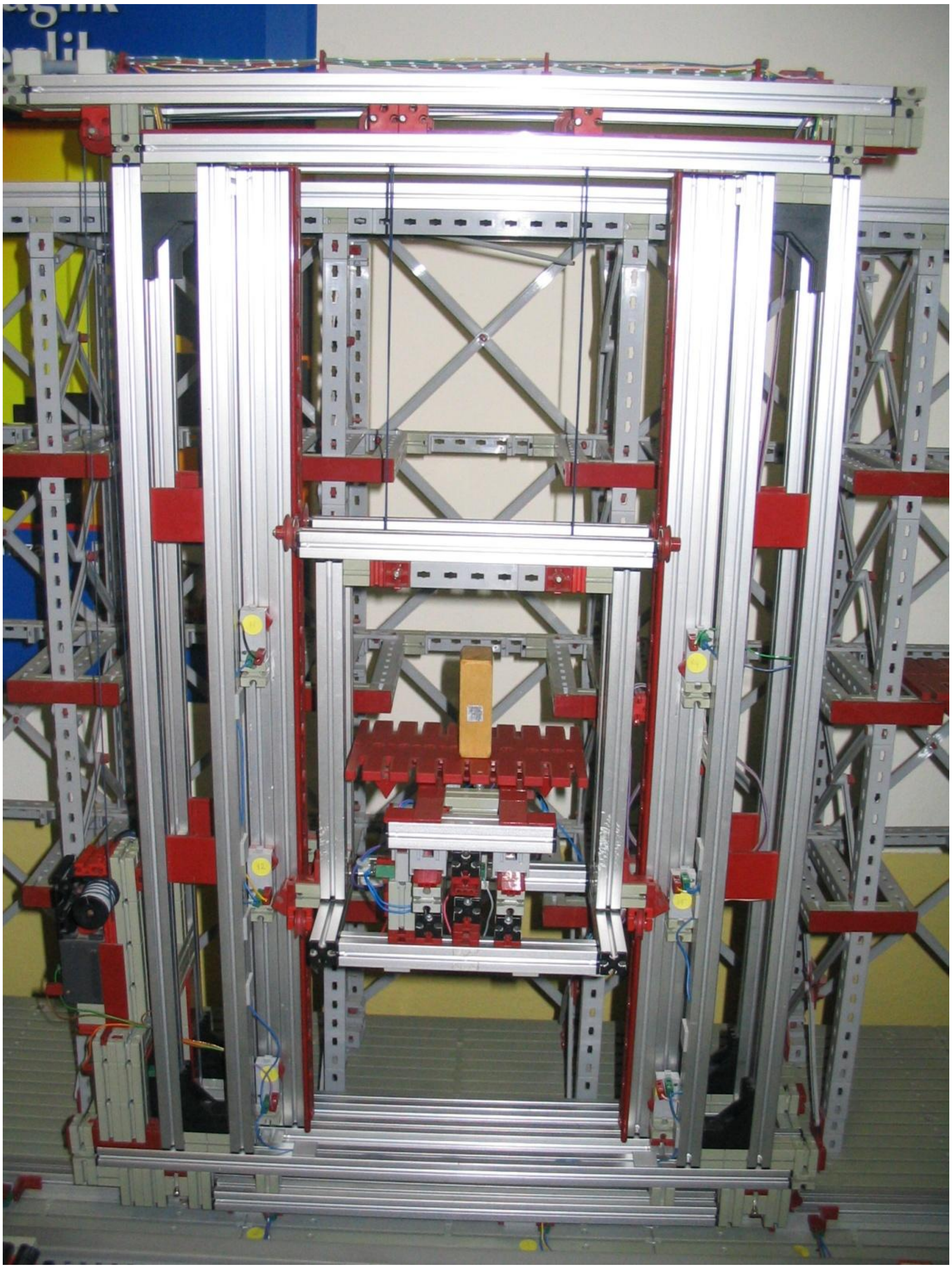


Figure A. 2: The stacker

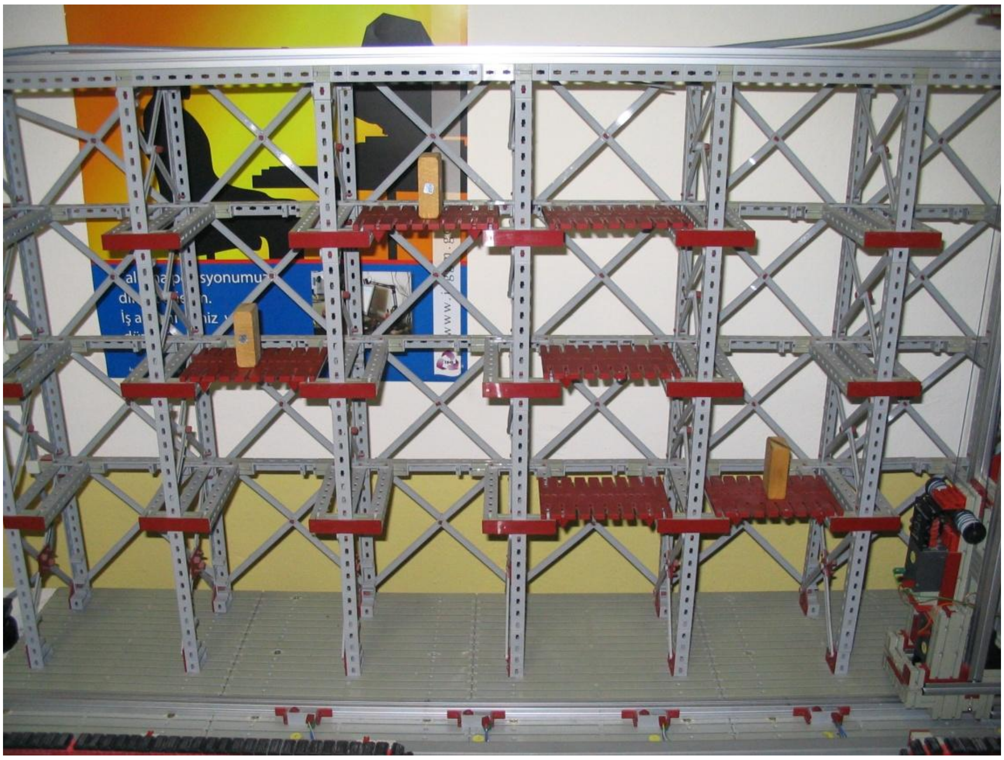


Figure A. 3: The rack and the aisle

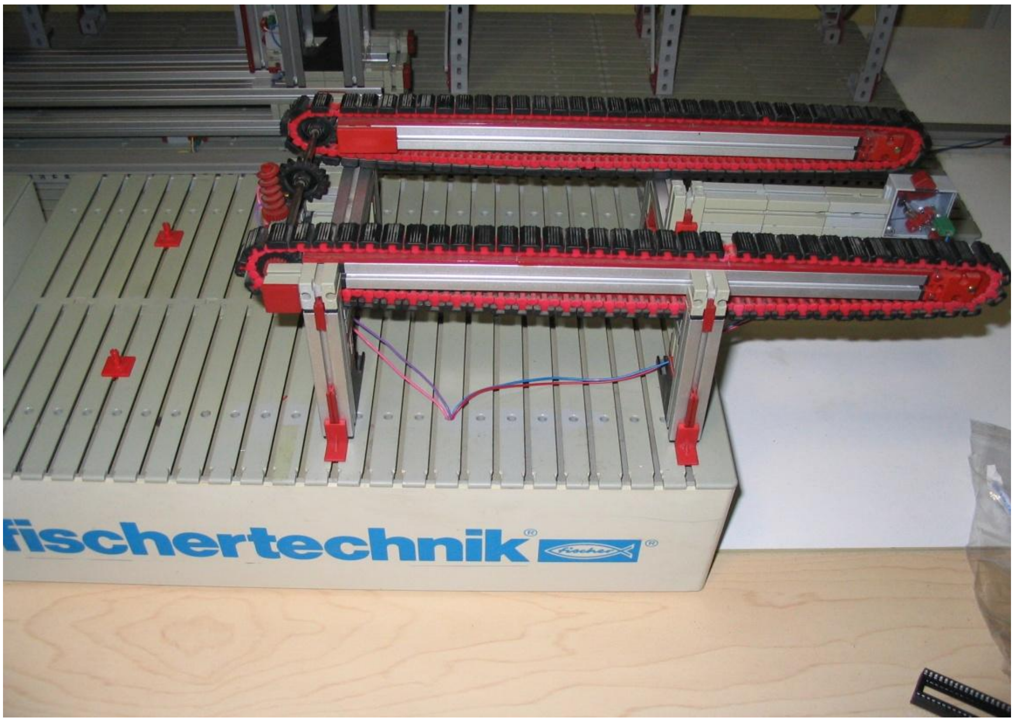


Figure A. 4: The input and the output pallet

APPENDIX B

USER MANUAL OF THE MINIATURIZED AS/RS

The Miniaturized AS/RS system mainly includes three parts as physical system, control boards and control software. The physical system consists of input and output pallet, robot arm, stacker and a rack. Equipments of the physical system have actuators and sensors. There are two boards in order to provide low level control in the system. One of these boards, which designated as Handling Board, controls the input pallet and the robot arm. The other, which designated as Stacker Board, controls the output pallet and the stacker. These boards communicate with a computer one at a time. There is software, which designated as Miniaturized AS/RS Control, runs on a computer and it controls the overall system.

B.1. Installing Boards

At this stage the Handling and Stacker Boards are interfaced with the computer and physical system. Each of these boards has RS-232 connector for communicating with the computer and digital input and output connectors for the actuators and sensors of the physical system. Also, there are five peripheral DC motor drivers which will be connected to these boards. There are a lot of digital input and output in the system so the cables are designated in order to prevent connection error. This designation is presented in Table B.1.

Table B. 1: Designation of the cables

Cable#	Description	Connected to
1	Reference button of the robot motor1	Handling Board
2	Reference button of the robot motor2	Handling Board
3	Reference button of the robot motor3	Handling Board
4	Reference button of the robot motor4	Handling Board
5	Input pallet button	Handling Board
6	Input pallet button	Handling Board
7	Gripper button	Handling Board
8	Common buttons of the robot arm	Handling Board
9	Power of the optical sensors	Handling Board
10	Ground of the optical sensors	Handling Board
11	Signal of the optical sensor1	Handling Board
12	Signal of the optical sensor2	Handling Board
13	Signal of the optical sensor3	Handling Board
14	Robot motor1	DC Motor Driver
15	Robot motor1	DC Motor Driver
16	Robot motor2	DC Motor Driver
17	Robot motor2	DC Motor Driver
18	Robot motor3	DC Motor Driver
19	Robot motor3	DC Motor Driver
20	Robot motor4	DC Motor Driver
21	Robot motor4	DC Motor Driver
22	Input pallet motor	DC Motor Driver
23	Input pallet motor	DC Motor Driver
24	Shuttle motor	DC Motor Driver
25	Shuttle motor	DC Motor Driver
26	Output pallet motor	DC Motor Driver
27	Output pallet motor	DC Motor Driver
28	Vertical motor of the stacker	DC Motor Driver
29	Vertical motor of the stacker	DC Motor Driver
30	Horizontal motor of the stacker	DC Motor Driver
31	Horizontal motor of the stacker	DC Motor Driver
32	Output pallet button	Stacker Board
33	Output pallet button	Stacker Board
34	Shuttle button	Stacker Board
35	Shuttle button	Stacker Board
36	Vertical button	Stacker Board
37	Vertical button	Stacker Board
38	Vertical button	Stacker Board

39	Vertical button	Stacker Board
40	Vertical button	Stacker Board
41	Vertical button	Stacker Board
42	Horizontal button	Stacker Board
43	Horizontal button	Stacker Board
44	Horizontal button	Stacker Board
45	Horizontal button	Stacker Board
46	Horizontal button	Stacker Board
47	Horizontal button	Stacker Board
48	Horizontal button	Stacker Board
49	Common button	Stacker Board

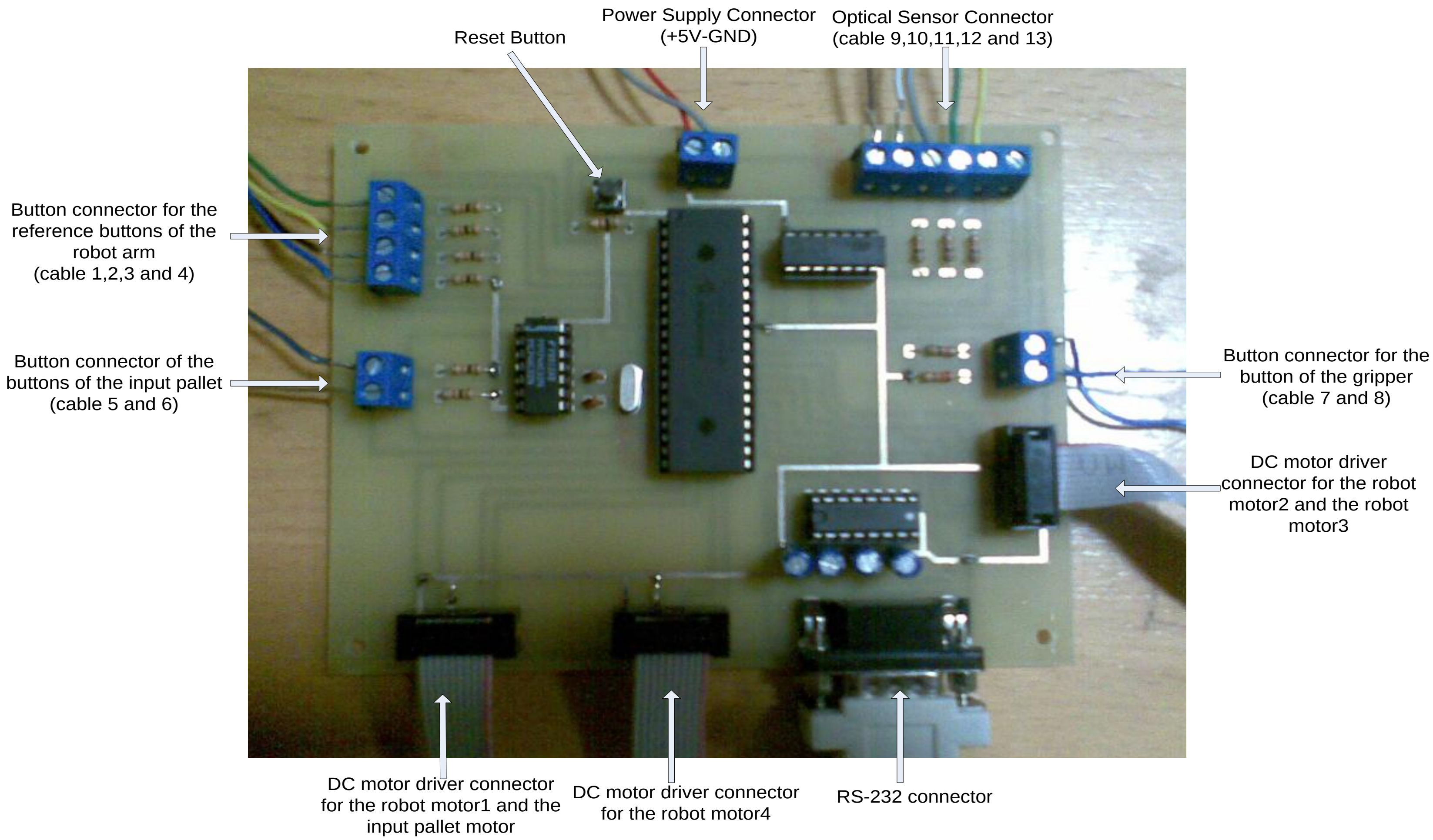


Figure B. 1: Connection details of the Handling Board

Connection procedure of the Handling Board is given below:

- Connect power supply to the Handling Board as shown in the Figure B.1.
- Connect the RS-232 cable to the board and connect the other end of the cable to COM1 of the computer.
- Connect cable 1, 2, 3 and 4 to the button connectors for the reference buttons of the robot arm as shown in the Figure B.1.
- Connect cable 5 and 6 to the button connectors for the buttons of the input pallet as shown in the Figure B.1.
- Connect ribbon cables to the 10 pin connectors for the DC motor driver. The DC motor driver is shown in the Figure B.2. There are 3 DC drivers for the Handling Board which drives 5 DC motors.
- Connect cable 7 and 8 to the button connector for the buttons of the gripper and common cable of the buttons as shown in the Figure B.1.
- Connect cable 9, 10, 11, 12 and 13 to the connector for the optical sensors as shown in the Figure B.1.

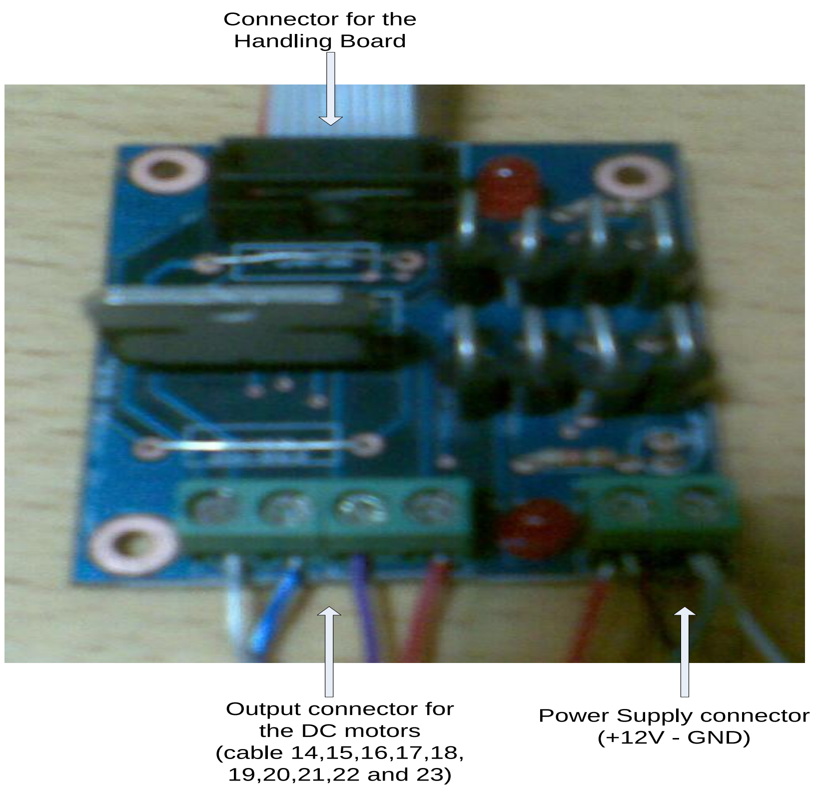


Figure B. 2: Connection details of the DC Motor Driver

Connection procedure of the DC motor driver is given below:

- Connect power supply to the DC motor driver as shown in the Figure B.2.
- Connect ribbon cable to the DC motor driver which the other end of it was connected to the Handling Board.
- Connect cable 14, 15, 22 and 23 to the first DC motor driver, connect cable 20, and 21 to the second DC motor driver and connect cable 16, 17, 18 and 19 to the third DC motor driver. Connection details for the output connectors of the DC motor drivers as shown in the Figure B.2.

Note: The Handling Board and the DC motor drivers are connected to different power supply unit. Ground of these power supply units should be connected.

After all the connections are made power supply of the Handling Board is opened and the firmware of this board is loaded to it. The firmware loading procedure is given below:

- Open the PIC downloader 1.08 and select hex file from the folder which includes the Handling firmware. User interface of the PIC downloader 1.08 is depicted in Figure B.3.
- Set port as COM1 and baud rate as 56000.
- Press Write button of the PIC downloader 1.08 and then press the reset button of the Handling Board.

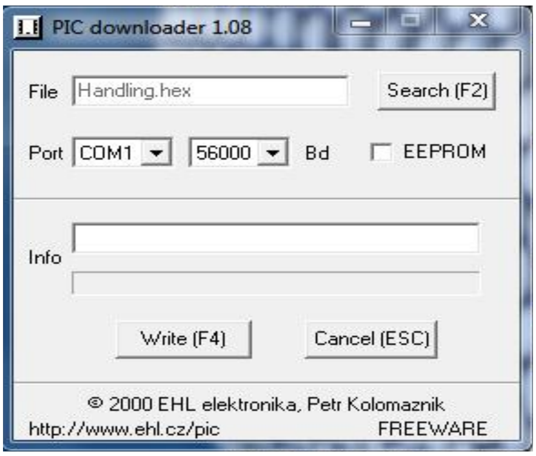


Figure B. 3: PIC downloader

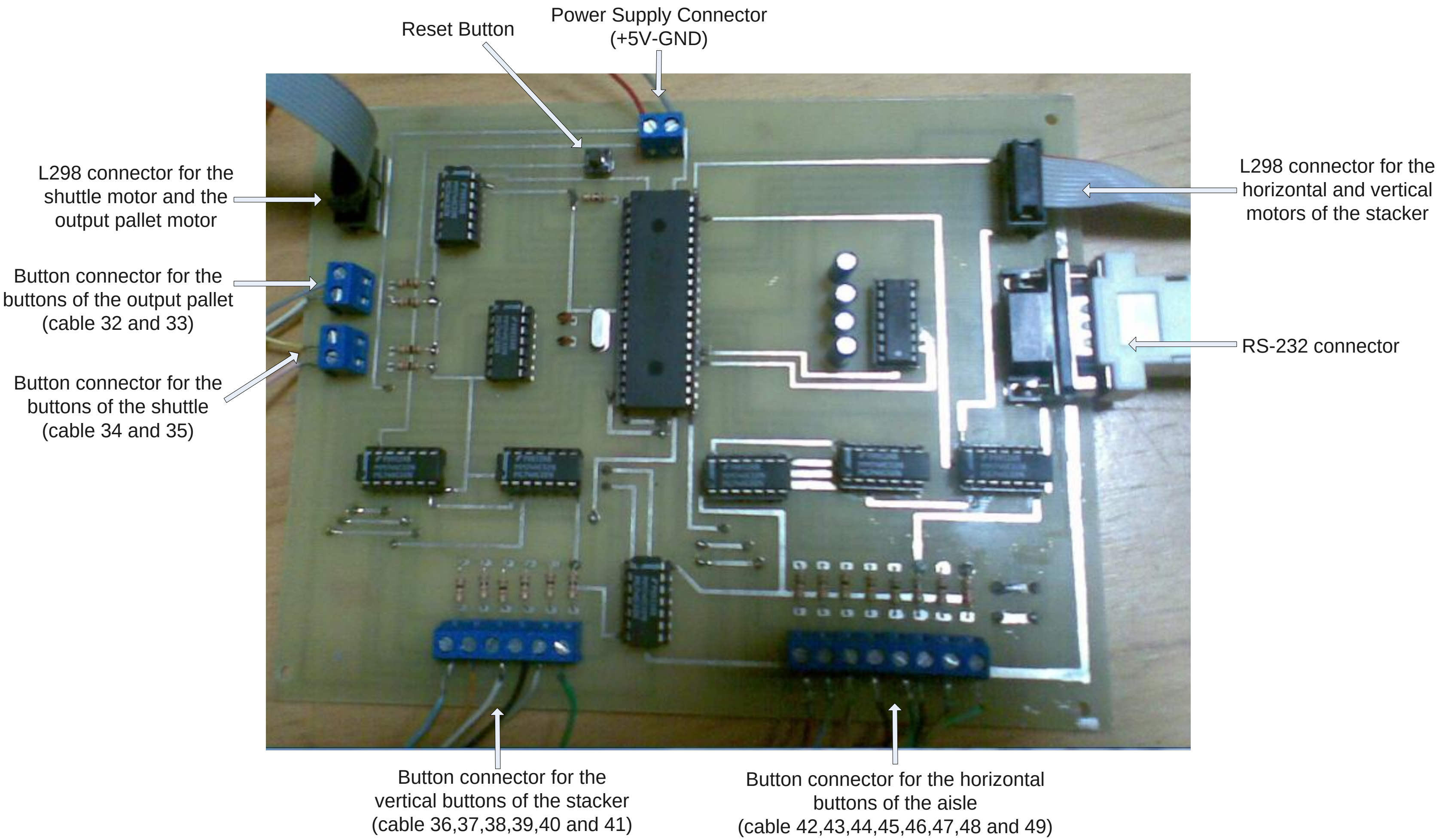


Figure B. 4: Connection details of the Stacker Board

Connection procedure of the Stacker Board is given below:

- Connect power supply to the Stacker Board as shown in the Figure B.4.
- Connect the RS-232 cable to the board and connect the other end of the cable to COM2 of the computer.
- Connect ribbon cables to the 10 pin connectors for the DC motor driver. The DC motor driver is shown in the Figure B.5. There are 2 DC drivers for the Handling Board which drives 4 DC motors.
- Connect cable 32 and 33 to the button connector for the buttons of the output pallet as shown in the Figure B.4.
- Connect cable 34 and 35 to the button connector for the buttons of the shuttle as shown in the Figure B.4.
- Connect cable 36, 37, 38, 39, 40 and 41 to the button connector for the vertical buttons of the stacker as shown in the Figure B.4.
- Connect cable 42, 43, 44, 45, 46, 47 and 48 to the button connector for the horizontal buttons of the stacker and connect cable 49 to the eight socket of the connector as shown in the Figure B.4.

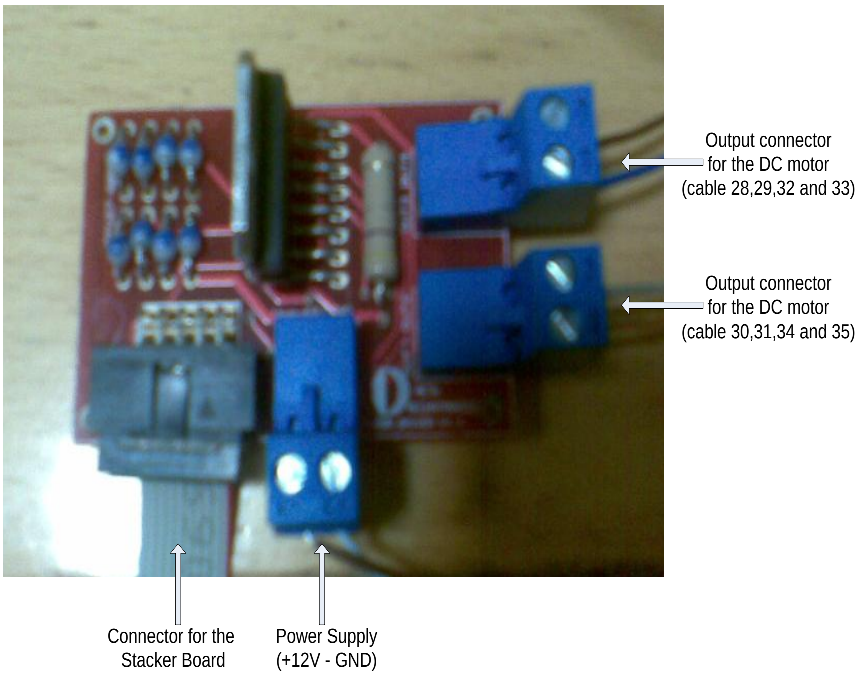


Figure B. 5: Connection details of the DC motor driver

Connection procedure of the DC motor driver is given below:

- Connect power supply to the DC motor driver as shown in the Figure B.5.
- Connect ribbon cable to the DC motor driver which the other end of it was connected to the Stacker Board.
- Connect cable 24, 25, 26 and 27 to the first DC motor driver and connect cable 28, 29, 30 and 31 to the second DC motor driver. Connection details for the output connectors of the DC motor drivers as shown in the Figure B.5.

Note: The Stacker Board and the DC motor drivers are connected to different power supply unit. Ground of these power supply units should be connected.

After all the connections are made power supply of the Stacker Board is opened and the firmware of this board is loaded to it. The firmware loading procedure is given below:

- Run the PIC downloader 1.08 and select hex file from the folder which includes the Stacker firmware.
- Set port as COM2 and baud rate as 56000.
- Press Write button of the PIC downloader 1.08 and then press the reset button of the Stacker Board.

B.2. Using Software

After all the connections are completed and the firmwares are loaded to the microcontrollers, the Miniaturized AS/RS Control is run. This program has a graphical user interface (GUI) and all the inputs of an experiment are introduced from it. Also outputs of an experiment displayed at the GUI and can be saved to text files. The GUI is presented in the Figure B.6.

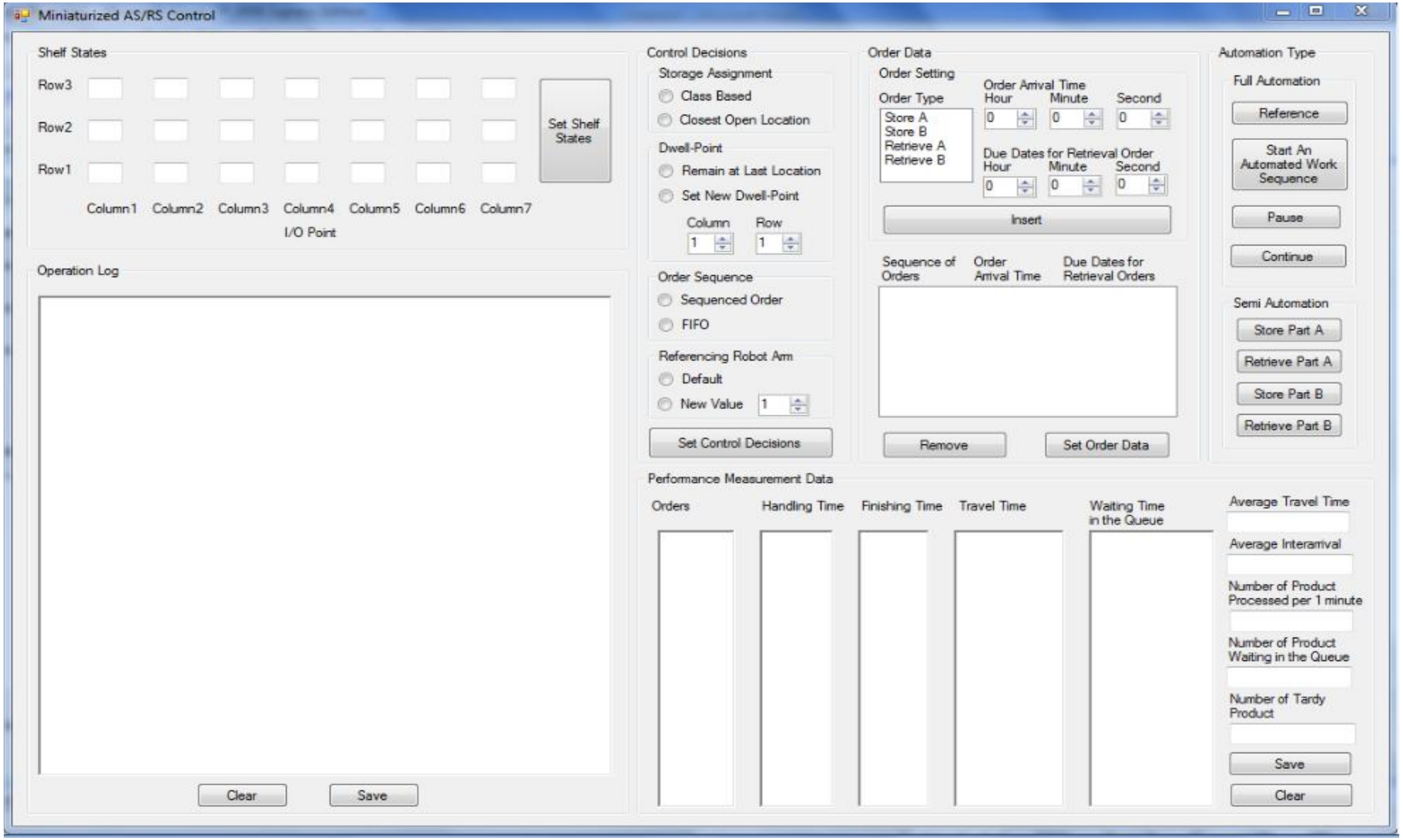


Figure B. 6: GUI of the Miniaturized AS/RS Control

Usage procedure of the Miniaturized AS/RS Control is given below:

- Write letters A or B or E or P to the textboxes which are in the Shelf State group control. These inputs are entered according to the state of the shelves. Then press Set Shelf State button.
- Select one of the radio buttons in the Storage Assignment group control.
- Select one of the radio buttons in the Dwell-Point group control. If the Set New Dwell-Point radio button is selected, define location from the column and the row numeric up down control.
- Select one of the radio buttons in the Order Sequence group control.
- Select one of the radio buttons in the Referencing Robot Arm group control. If the Set New Value radio button is selected, define value from the numeric up down control. The default value is four.
- Press Set Control Decisions button.
- Select order type from the list box in the Order Setting group control and set arrival time and due date of this order. Then press Insert button in order to constitute an order sequence.
- If there is a wrong order in the Sequence of Orders list box, select this order and press Remove button.
- If an order sequence is completed, press Set Order Data button.
- Press Reference button and wait until the physical system reaches to the reference position.
- Press Start an Automated Work Sequence button in the Automation Type group control.
- After completion of an order sequence, press Save buttons, those are in the Performance Measurement Data and Operation Log group controls, in order to record performance data and operation log of the operation.
- Press Clear button for clearing the textboxes in the Performance Measurement Data and Operation Log group controls.