23

Modelling Mechanical Systems

By
A. Levin

A thesis submitted in partial fulfilment of the requirement for
The Degree of Master of Industrial Administration

Department of Mechanical Engineering
University of Cape Town
Rondebosch 7700
South Africa

September 1999

Declaration

I the undersigned hereby declare that the work contained in this thesis is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.

.... Date..

Ari Levin

Table of Contents

List of Illustrations

Acknowledgements

I wish to express my appreciation to the following people:

Prof. G Nurick for his continual guidance, enthusiasm and support throughout this project and for allowing me to fulfil a childhood dream of being left alone for a year in a room full of Lego.

Prof. T Ryan for his guidance and teachings into Paradigm and Systems Thinking as well as the many obscure games and programmes which added another perspective to life and the environment, and improving the way I view, process, understand and react to the changing environment in which we live.

The workshop staff for assisting me with the construction of various non-Lego parts.

Ms E Warner of Lego SA for providing a large amount of Lego.

Mr K William's of Festo SA for donating the various pneumatic components.

Mr J Meyer for his assistance with the electronic circuitry used in this project.

Mr BI Reid and Mr D Finders for the use of their video editing facilities.

Synopsis

Education should be one of the highest priorities of any country. The ability to pass on knowledge and understanding effectively is a sign of a good educational system. From an engineering perspective, better teaching tools allow for greater understanding by the students making them better engineers.

Educational aids have been with us for a long time. Inventions from the abacus to the personal computer, have aided engineering students in faster understanding and application of their work. These aids are costly and sometimes become affordable only when outdated and redundant. A low-cost device that does not date would be the educationalist's ideal solution.

To this end, a concept model of a conveyor sorting system was developed using Lego as the design medium. The model is a conveyor belt along which coloured blocks are sorted into bins by pneumatic cylinders. The model is controlled by a personal computer and the Lego interface box. The software is written in Visual Basic.

Objective: To investigate the viability of Lego:

1. As an educational aid in the engineering curriculum.
2. As a design medium for modelling real world problems.
3. By reviewing literature of educational philosophies and practices, particularly current uses of Lego in educational environments in order to evaluate its potential.

Results:

1. The model was tested and found to have potential as a valuable teaching aid. It was concluded that Lego could be used successfully as a modelling medium for any mechanised system requiring controlled movement. It can also be effective on large scale models of production lines.

At this point, it must be noted that between the model work and submission of this project several successful projects based on this work were completed between the end of the model building and the submission of this document, these include:

1. Hrabar S.E, "Automation Of A Warehouse Packing And Inventory Control", Postgraduate Thesis, Dept. Mechanical Engineering, UCT, November 1996.
2. Tollner M.E, "Modelling Of Mechanical Systems Using Computer Interactive Devices", Undergraduate Thesis, Dept. Mechanical Engineering, UCT, Nov, 1995.
3. Tollner M, "Designing And Modelling An Automated Packaging System", Postgraduate Thesis, Dept. Mechanical Engineering, UCT. 1997.
4. Kotze, J. "A Concept Model For A Multi-Fingered Prosthetic Hand", Postgraduate Thesis, Dept. Mechanical Engineering, UCT, 1997.

The objectives of this project were found to be viable and at the conclusion of the project recommendations were made. These recommendations, however, have been incorporated into the proceeding projects and making recommendations at this time could be construed as hindsight.

Chapter 1. Introduction

1.1 Models

The education of future generations of engineers is a serious undertaking, yet educationalists in this field have always been conservative about the introduction of new models in the curriculum. Models of all sorts have a beneficial role in the education process.

There are many types of models used in any engineering course. The two basic types are: physical and graphic. Physical models are real representations of an actual system albeit that the model may be a scaled down or non-functioning version. This model may or may not be constructed of the same material as the item being represented. Graphic or numerical models simulate an occurrence mathematically, visually or both. These can be represented by a flow chart on paper or graphically on a computer screen. In the engineering curriculum models are used to represent the real world and as such modelling mediums, particularly in the physical category, have to be constructed of real world materials in order to convey the gravity of the subject.

The development of models has been boosted by the development of the computer. Prior to the computer all models were either physical or numerical, from concepts of bronze statues to buildings and aircraft. The models consisted of a series of drawings or schematics and possibly a scaled down physical representation of the finished product. This was of course so that the client could have a better idea of their finished product. With the development of numerical modelling during World War II and later the development of the computer, physical models, although still widely used, began to take a back seat to numerical models. Buildings, motorcars and aeroplanes were designed and tested using physical models, but as computing power and processor speed increased, this design and testing was being achieved numerically rather then physically. Numerical design and testing on aeroplanes for instance, has virtually made physical testing obsolete. All testing can now be done virtually. Where in the past many hours were spent testing the aerodynamics of a model aeroplane in a wind tunnel, now this can all be accomplished in a fraction of the time and cost with a computer.

For education and industry this is a major boon and of great importance, but in order to take advantage of technology one has to have the financial resources to afford. Computer speed and technology are progressing at such a rapid pace that state of the art computers are becoming out dated in short periods of time. Only companies that generate large revenues can afford to keep pace with this technology. Educational institutes, particularly in South Africa, do not generate sufficient revenue and rely heavily on government subsidy and donations from the private sector. This income does not allow these institutions to maintain the same technological edge as in the private sector.

Physical modelling, although in decreasing use in industry is still a valuable and much used tool in engineering education. It is widely used for teaching the basics of mechanics and mechanical systems. These models do not date or need replacement. New models need only be constructed to model new mechanical systems. The material from which these new models are constructed is the focuses of this project.

The objective of this project is to develop a concept model of a conveyor system. The development of a more sophisticated representation of an industrial system is a long term goal, but beyond the scope of this project. A concept model has to be developed in order to evaluate the mechanisms, control and the building medium prior to attaining the ultimate goal. The medium for the concept model in this project is Lego. The decision to use Lego was based on the large variety of components in the Lego range and to validate the secondary objective of this project, to evaluate Lego as an educational tool. Lego presents the designer with a large selection of components and mechanical elements that can be assembled and dismantled with ease. Tailoring Lego components to a specific shape and size is easy as they are made of plastic. Two more elements have been added to the Lego range, Lego Technic and Lego Dacta. Lego Technic has an assortment of gears, shafts, pulleys and conveyor belt elements as well as pneumatic cylinders and pumps. Lego Dacta has various electrical and electronic devices, such as motors, lights and sound elements as well as temperature, angle and light sensors all controlled through an interface connected to a personal computer.

For this project a study of educational trends, the use of Lego in education and the development of physical and numerical modelling using Lego was undertaken. This literature survey will provide the starting point of integrating Lego into education as a modelling medium.

The aim of this project is not to construct a scaled production line able to manufacture a product, but to develop the concepts needed in order to achieve such a model in the future. The aim of the concept model is to test the building material, Lego as well as the limitations of mechanisms constructed out of it. It would be unrealistic to expect a Lego model to perform at the same levels as the system it is simulating. The importance here is to determine whether Lego has the potential to be used as a modelling medium and an educational tool in the future.

The control of the conveyor sorting system is through a personal computer and the software written in Visual Basic. The computer receives all information from the various sensors and through the software activates and controls the motors and actuators. The actuators are pneumatic cylinders that are extended and retracted through electro-pneumatic switches. The software has graphical readouts that show the workings of the software.

Chapter 2. Literature Review

2.1 Education

"Now, what I want is, facts. Teach these boys and girls nothing but facts. Facts alone are wanted in life. Plant nothing else, and root out everything else. You can only form the minds of reasoning animals on facts: nothing else will be of service to them..." The speaker and schoolmaster {Mr. Gradgrind} and a third grown person present, all backed a little, and swept their eyes on the inclined plane of little vessels then and there arranged in order, ready to have imperial gallons of facts poured into them until they were full to the brim." [1:Dickens, 1954].

This is one view of education, although a harsh example "Synchronisation and Standardisation" from a "Second Wave" society, [2:Toffler, 1980] (A first wave society is defined as the agricultural revolution where people settled onto a fixed area of land. The Second Wave is defined as the Industrial revolution where people moved off the farms and into the cities) where children in an industrial culture were conditioned from an early age to be good factory workers. School, just like the factory was a nine to five "job", reading, writing and arithmetic constituted all the facts an industrialised child would need in order to fit into his or her role as a factory worker. This education was designed to produce good factory workers who's sole requirements to be good factory workers were punctuality, obedience and rote. (It is interesting to note that pre-industrialised "First Wave" society which is defined as an agricultural based society, where time is measured by the changing seasons. Yet in the Second Wave time is measured by the watch. Farmers rise with the sun and plant and harvest depending on the season).

This philosophy of education has been significantly modified to fit the needs of a "Third Wave Society" (Third Wave is a move from an industrial base to an information age) [2:Toffler, 1980]. In this modification, there is a move to equate education with fun. In a report the US Secretary of Education describes elementary school science as "... a grab bag of facts and stunts " and concluded that "...seen only as a laundry list of theories in a workbook, science can be a bore." [3:Bennet, 1986]

One solution put forward in "Mindstorms", [4:Pappert, 1980], is to create "Mathland", a classroom, where children could learn mathematics as a "living language". The example is given of learning French in France rather than in an American classroom. Some schools have seen the need for change and adopted a "hands on approach", [4:Pappert, 1980], to teaching. In science for example, students would be taught Newton's Second Law : F= ma and its application. This would have been where Mr Gradgrind's "vessels" would have stopped. The "hands on approach" would now allow students to conduct an experiment to prove this law.

According to a US Department of Education report, this "hands on approach" is however only an improvement, but not much else. These are merely activities of "recreating" someone else's experiments and do not better educate the student. The students are told what to measure and from the theory know what the answer should be. "This approach to science continues through school and into universities. Students have often written up laboratory reports by calculating backwards"[3:Bennet, 1986]. That is, they start with the right answer and work backwards to the question. The US Department of Education report concluded that "students learn from this exercise, but certainly not science." [3:Bennet, 1986].
In "Mindstorms" [4:Pappert, 1980], the theory is put forward that children learn best when they are actively involved in creating and constructing meaningful products. This approach is called *constructionism:*

> "First, the contructionist approach creates an environment in which students act like "real" scientists and "real" inventors. As a result, students are in much closer contact with the important ideas of science. Students do not simply learn facts, they learn a way of thinking critically and systematically about problems and even about the process of solving them.
> Second, the approach creates an environment in which students take pride in their work. As long as science and mathematics are viewed as a collection of formulae and rules, students will never really know or understand mathematics or science." [4:Pappert, 1980].

This approach has been used in "Introduction to Technology" workshops run by the South African Institution of Mechanical Engineering (SAIMechE) and described as a "valuable tool in SAIMechE's plan to spread technology awareness to school children" [5:DeSousa, 1992]. This programme also boosted interest to continue students education in the field of science and engineering. It introduced and explained mechanical concepts such as gear ratios and velocities, levers and pulleys. The students were divided into groups of three or four and assigned a project. These projects, for example: constructing a self propelled car, a crane or a clock's gear mechanism using the principles they had learned. They were given a box of Lego, adhesive tape, string, springs and elastic bands and the freedom to complete their project.

In this environment, each group designs and constructs their assignment based on their interpretation of how the their model should be built. A self-propelled car can have many power sources and it is the group's decision as to which will work best, based on their intuition and testing. Realising that there are many solutions to a problem with no best solution it is up to the group to decide, based on the components at their disposal, which solution to use. The concluding remarks by the students are evidence of the learning process they had undergone.

Using this same constructionist approach, Martin and Resnick set an exercise dubbed "The Soap Box Derby" [6:Martin,Resnick, 1993]. The goal was to construct a motorless car that would go the furthest distance down an inclined ramp. In this exercise school pupils had to put forward theories regarding factors which would influence the car's behaviour. Theories such as size of car, size of wheels and mass of car were thought to have an influence. Cars based on these theories were built and then tested. During testing the distance achieved by the car was noted. Students learnt about measurements, some who had not learned to use a ruler devised other units of measure such as book lengths. They also realised that the same car did not go the same distance each time, thus the idea of averaging arose naturally. Some students recognised the advantage of using numerous trial runs to calculate the average.

In another exercise to give students an idea of how "real inventors" work, they were shown copies of actual patents. Students were then encouraged to submit their own Lego/Logo patents. Patents that were submitted included: a sewing machine by a fourth grade girl, a rack-and-pinion steering mechanism by a fifth grade girl and a third grade class (with outside help) invented a way to use Lego motors as generators to turn on Lego lights [6:Martin Resnick, 1993].
The findings from these exercises were that projects often started out in one direction and then shifted to a new direction when something went "wrong". Through these experiences, students learned an important lesson about invention and design. The conclusion from these exercises was "Invention is not like following a recipe from a cookbook. It is full of trial and error, educated guesses and serendipity. Students, like real inventors, learned to keep their eyes open for the unexpected." [6:Martin Resnick, 1993].

The constructionist approach to education requires a teaching medium in order for the students to achieve their goals. In the "Soap Box Derby" [6:Martin Resnick, 1993], Lego is that teaching medium. Lego, although viewed as a child's toy, is essentially brightly coloured plastic blocks, that come in different shape and sizes. Blocks that are usually associated with children playing "make believe", now have another use, Education. Lego has been successfully used at the University of Cape Town (UCT) as well as at the Massachusetts Institute of Technology [7:Martin, 1992] in their engineering curriculum.

The Academic Support Programme for Engineering in Cape Town (ASPECT) [8:Sass, 1988] at UCT aims to bridge the gap between secondary and tertiary education of disadvantaged students. At the "Introduction to Mechanics" Workshop [9:Jawitz, 1992], one module of the ASPECT course, a class of 60 students was divided into 20 groups of 3 students each. The workshop could only accommodate 10 groups so each student attended a 1½ hour workshop every fortnight. The list of projects appears in Appendix 2.

In the first cycle the groups were given Lego kits complete with assembly manuals and were required to construct the model in a 1½ hour session. This was done to familiarise the students with the Lego systems both electrical and pneumatic. In the second cycle students were given the Lego picture of the model to be built without assembly manuals and given several sessions to complete the task. The third cycle was run as a competition with a prize for the winning team. Students were not given any plans or diagrams and had to design and construct working of their own choice within one hour. Following the course a questionnaire was given to the students, whose response was "overwhelmingly" positive [9:Jawitz, Nov. 1992].

> *"It helps to explain some of the questions and problems we did in Applied Maths to see how to put them into practice."*

> *"It teaches you how to design a model and actually build what you have designed, also it helps to know how machines in industries work and how they are built."*

> *"It gives a student an idea of what Engineering is about."*

There were, however those who felt that the use of Lego had undermined the seriousness of the course.

> *"We should get serious stuff, not plastic models that's not real."*

The comments from the lecturers were that the course helped students develop ways of working in groups and provided them with the opportunity to develop their English language skills especially with respect to technical vocabulary, reading and writing skills.

The use of Lego was introduced into the second year design course in the Department of Mechanical Engineering. The aim of the Lego component was to familiarise the students with team work as well as engineering design and technical communication.

The students were divided into 6 groups of 5-6 students. Each group was assigned a problem sheet (see Appendix 1). Each group was given a 3 hour session in which to design and construct a working model to solve their particular problem. They were then required to present their Project the following week, comprising of : a solution specification, a technical drawing and a parts list. This Project was then handed to another group who were required to construct the model based solely on the design given to them.

Once the models were completed, each group constructing a model graded the design based on the ability to construct the model from the technical drawing and parts list and whether the design solved the original problem. Finally other groups gave input into how better to solve the problem.

The students comments showed that they learned more from the discussion that followed , than from the designing process i.e. construction and drawing up of the Solution Specification. There were however comments from the students that the use of Lego was inappropriate as it imposed limitations on what could be constructed based on the parts available. This point of view was changed later when the limitations of materials such as steel and concrete were discussed.

2.2 Models

"A three dimensional drawing on a two dimensional screen can still be interpreted in more than one way. But an accurate scale model can settle most arguments that arise." [10:Rowland, 1986]. This statement although true, was made in 1986. Since then the advancement of computer hardware and software technology and in particular the Computer Aided Design/Computer Aided Manufacturing (CAD/CAM) developed by the National Aeronautics and Space Administration (NASA), has allowed Gulfstream Aerospace and Boeing Corporation to design and electronically assemble their Gulfstream V corporate jet and Boeing 777 jet in a virtual hanger [11:CNN, 1996]. The Gulfstream V and Boeing 777, were never wind tunnel tested and no prototypes were ever built and tested. The first plane to be built was the final product. All the testing, plant layout and assembly sequence were conducted in virtual reality.

A simulator, however, does not need to be as complex as the NASA CAD/CAM simulator. A simple simulator such as "The Beer Game" [12:Senge, 1990] can teach valuable lessons about stock ordering and sales forecasting.

Over the past two decades , the manufacturing industry world-wide has undergone radical and extensive changes turning production capability into a competitive edge. These changes have involved a shift towards advanced technologies such as computer aided design and manufacturing (CAD/CAM), Materials Resource Planning (MRP), Just in time (JIT) and total quality circles philosophies [13:Chase, Aquino, 1989]. The new generation of manufacturing systems have high levels of integration and flexibility. These systems are capable of performing production functions that are interrelated, resulting in extremely complex systems. The advanced nature of these systems relies on the sound integration of equipment, methods, monitoring and control, critical for operational and financial success. There are always a number of configurations that are technically feasible for each manufacturing system. In order to keep a manufacturing system efficient and competitive, the optimal configuration must be in operation and for the system to keep up with product development, equipment changes and changing operating expenses, its configuration must be constantly reviewed [13:Chase, Aquino, 1989]. Each configuration has different operating parameters with respect to product range, processing sequences, machining systems, materials handling systems, operating logic, expansion opportunities and a host of other parameters.

Determining the optimal system configuration requires planning based on information about the system. In most situations considering the entire system as an operating unit is impossible due to the system's complexity or interrelationships, so planning decisions are based on a combination of the system's elements operating characteristics, with their interacting characteristics neglected. This necessitates the application of a modelling technique that simulates the performance of available alternative configurations and that forms a coherent approach for obtaining the most suitable strategy to follow.

Simulation predicts the characteristic behaviour of complex manufacturing systems by calculating the actions of and reactions between the system elements. By following the process through the system's work stations and by examining the conflicting demands for limited resources, the planner can make intelligent decisions on capital investment planning, physical layouts, labour utilisation, quality control methods evaluation, bottleneck identification and removal, inventory or buffer utilisation and other factors concerning the system [13:Chase, Aquino, 1989].

2.3 Computer Simulation

Computer simulation is one of the most powerful tools of analysis available to design makers regarding the design and operation of complex processes or systems. Results from a simulation provides a greater depth of information on which to base decisions. In an increasingly open international market where operating expenses are rising, systems are becoming more complex and intricate, customers are more demanding and product and service quality more important and competition fiercer, simulation has become an essential, indispensable decision making tool for engineers, designers and managers. With a simulation, one has the opportunity to evaluate the practical alternatives available either in support of strategic initiatives or in the continuous search for better performance at the operational and tactical level. This can be done without experiencing the enormous expenses and difficulties associated with implementing changes in an operation, especially when the ultimate benefits are in doubt.

2.4 Systems

The object to be modelled must be viewed as a *system*; this is crucial to systems thinking [14:Ackoff, 1984]. The systems approach considers the total system's performance rather than concentrating on the elements that make up the system. The essence of the systems *approach* is that in order to maximise the output or productivity of the system, each individual element making up the system need not be functioning at its optimum rate [14:Ackoff, 1984].

2.5 Continuations

Several years have past between the completion of the work on this project and the finalisation of this report. During this time the ideas presented in this report have been advanced by several students who have verified the use of Lego in both the study of modelling systems.

Hrabar S.E, "Automation Of A Warehouse Packing And Inventory Control", this thesis developed a system that packs and unpacks a warehouse automatically, while keeping record of the stock [16: Hrabar, 1996].

Tollner M.E, "Modelling Of Mechanical Systems Using Computer Interactive Devices", this thesis developed a model of a fully automated factory packaging system [17: Tollner, 1995].

Tollner M, "Designing And Modelling An Automated Packaging System", this thesis developed an improved materials handling system for a beverage cap production factory [18: Tollner, 1997].

Kotze, J. "A Concept Model For A Multi-Fingered Prosthetic Hand", this thesis designed a computer controlled and operated prosthetic hand [19: Kotze, 1997].

Chapter 3 Development of the Model

3.1 Introduction

A concept for a conveyor sorting system (CSS) model first was developed. The model of the conveyor sorting system is constructed of Lego and controlled by a personal computer. The software developed to interface with the computer is graphical and provides easy control of the model by means of several buttons. It can be run in automatic or semi-automatic mode. In automatic mode, the model automatically releases the blocks and sorts them into their respective bins (blue and white). In semi-automatic mode, the rate of release of the blocks onto the conveyor belt can be manually operated and the blocks have to be manually sorted into their respective bins (blue and white).

The only function that is completely manual is the loading of the blue and white blocks into the magazine. Once the magazine is loaded and the model started, the feeder mechanism loads blocks from the magazine onto the conveyor belt. As this is being done, the light sensor is activated and through the software the colour of the block is determined. Once the block is on the conveyor belt, it is moved along until it reaches the correct pusher arm, which is then activated and pushes the block off the conveyor belt and into the bin.

The model is made up of one Lego 9V motor (Figure 1), four Lego pneumatic (Figure 2) cylinders. The sensors in the CSS are one Lego angle (Figure 3), one light(Figure 3) and one touch (Figures 4) sensor. The body or structure of the CSS is constructed from standard Lego components as in Appendix 3.

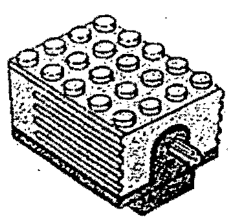


Figure 1 Lego Motor.

Figure 2 Lego Pneumatic Cylinder.

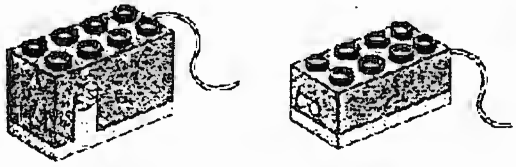


Figure 3 Lego Angle and Light Sensor (L-R).

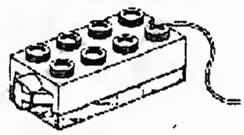


Figure 4 Lego touch sensor.

The conveyor belt is operated by the Lego 9V motor through a gearbox (Figure 5). The pneumatic cylinders, are operated by a pressurised air cylinder (Figure 6) which is operated via electromechanical switches, (Figure 7) which are in turn controlled by the Lego interface (Figure 8) via relays (Figure 9).

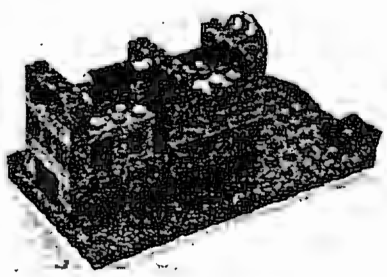


Figure 5 Lego Gear Box.

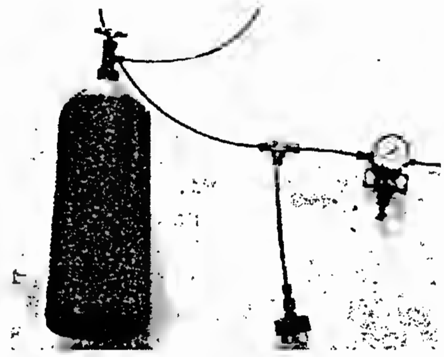


Figure 6 Air Cylinder, Pressure Gauge and Pressure Reducer.

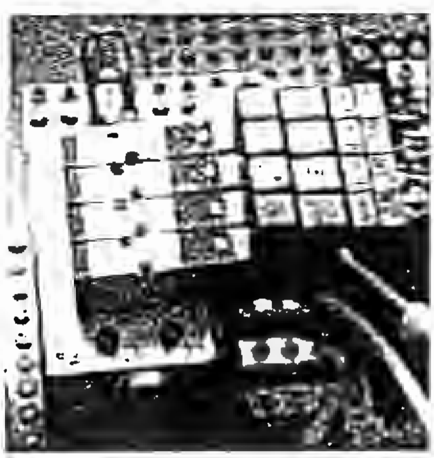


Figure 7 Electromechanical Switches.

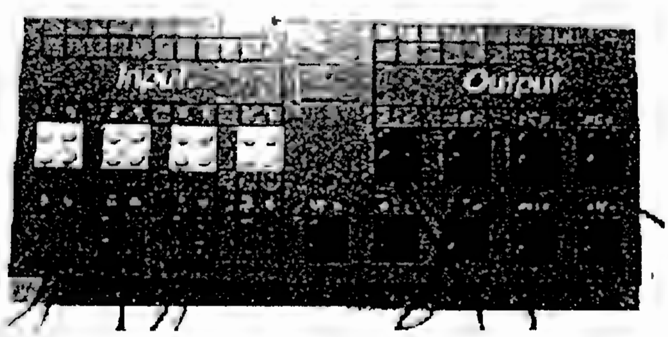


Figure 8 Lego Interface.

The software tracks the movement of each block along the conveyor belt. This tracking is achieved by an angle sensor linked to the conveyor belt which measures the distance travelled along the conveyor belt (Measured Angle x Cog Radius). Should the conveyor belt stop, slow down or speed up, the tracking of the blocks is unaffected.

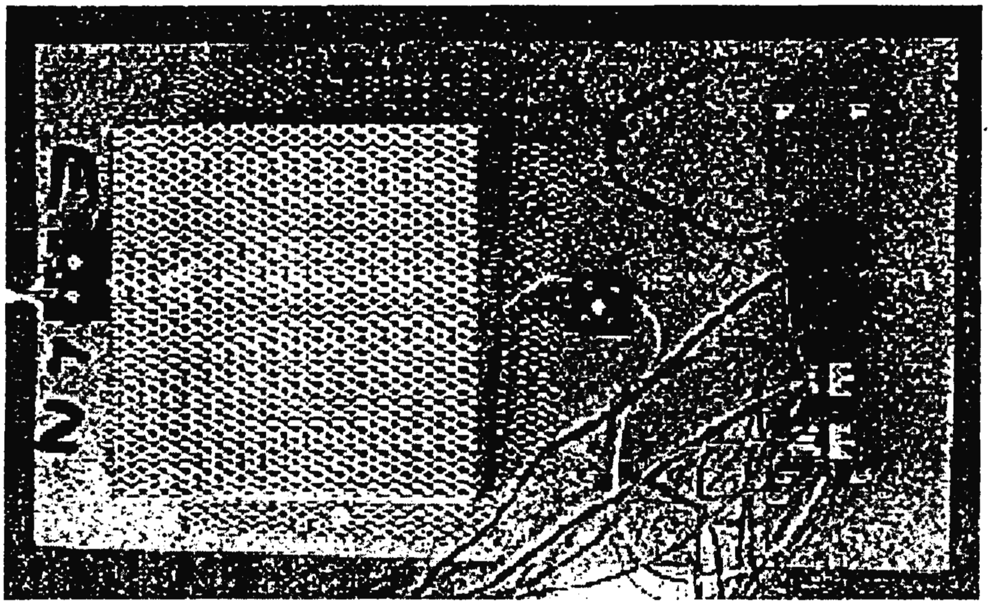


Figure 9 Transformer and Relays.

3.2 Initial Decisions

The initial decisions of the conveyor sorting system were to place boundaries on the scope of the project. The use of Lego as the building medium was a prerequisite as it was integral to the objectives, i.e. to determine its usefulness and ability to construct models. Additionally the Lego was being assessed for its efficiency as an educational tool in graduate and post graduate studies. As a result of this work a Lego laboratory was set up.

Due to the initial small quantity of Lego available, it was decided that the model would be small. There would be four minimum requirements in order for the model to achieve its function;

1. Manipulate Objects
2. Identify objects.
3. Move objects
4. Automate and control the above activities.

These were applied to the CSS in the following way:

1. loading blocks onto a conveyor belt.
2. Identifying the block colour.
3. Moving the blocks along the conveyor belt.
4. Moving the blocks off the conveyor belt and into a bin.

It was decided that only two colours would be used in the model.

3.3 Design Methodology

The design methodology of the Conveyor sorting system was of an empirical nature. The process of which can be seen in Figure 10.

The design process began with a goal, to construct a working Lego model that would incorporate the following functions:

1. Manipulate Objects
2. Identify objects.
3. Move objects
4. Automate and control the above activities.

These decisions would be conceptualized, constructed, tested and evaluated. The design would then be modified, accepted or rejected.

Lego is a very flexible working medium. The components are off the shelf and cheap. This allowed for the design process to be iterative as mistakes would not be costly. This in itself is an advantage for the learning environment. Each idea to solve a particular problem could be immediately designed, tested and evaluated.

Each model element comprised two parts, the Lego model and the control software which drives the Lego model component. After all the model elements were accepted they were integrated into a single system. Each model element as well as its software had to be modified. Following each modification the system was tested and evaluated.

The model was complete when all the elements were integrated into a single working system.

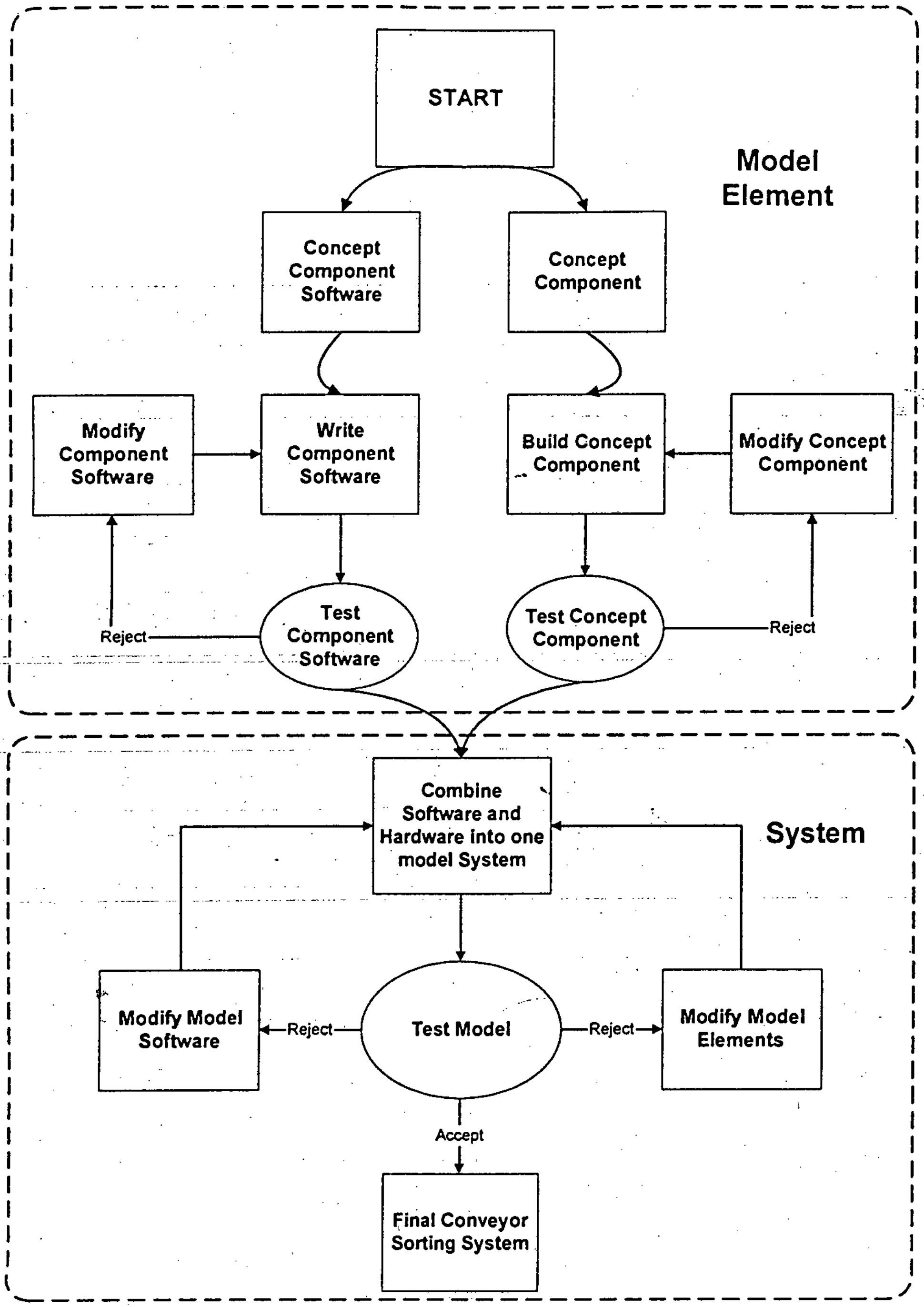


Figure 10 Computer Model.

A literature review was carried out on modern trends in education, Lego and its uses in education and models and applications of models in simulation.

Using Lego as a building medium is ideal for this methodology as the mechanisms can be built quickly from concept. The mechanism can be tested and evaluated and changes can be made easily. The small forces produced by the actuators made structural analysis unnecessary. Power calculations for the motors were not undertaken as the system is not dependent on constant speed.

3.4 Actual Design of the Conveyor Sorting System

3.4.1 Initial Design Parameters

From the Lego components available it was decided that the length of the conveyor belt would be the maximum length possible with the available conveyor belt link components (Part No. 9852, Appendix 3).

The blocks to be sorted were initially flat 1x2x1 (length, width, height, flat top surface) (Figure 11).

These were later changed to 2x2x1 in the final design as the smaller blocks caused the system to jam. Flat blocks were chosen as the surface would be better suited for the light sensor to detect the block's colour. The flat surface made stacking of the blocks easy without the blocks "locking" together. It also made pushing the block through a rectangular slot easier than with ridged blocks.

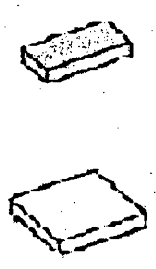


Figure 11 Lego Tiles: 1x2x1 (Above), 2x2x1 (Below).

3.4.2 Actuator Sources

A design criteria was to use Lego components where possible, unless Lego components were unsuitable due to design constraints.

Using Lego as the building medium gave two choices for actuators: An electric motor and pneumatic cylinders. Since the model to be constructed was not to scale, the size of the components was not important. The electric motor is powered either through a Lego battery pack or through the Lego interface box, (Figure 8). The pneumatic two way cylinders have a manual piston pump and manual switch (Figure 12). The pump and switch were found to be unsuitable as the pressure supplied by the pump was not sufficient to extend or retract the piston to its maximum position and the switch had to be manually operated. Secondly the pump and switches could not be integrated into an automated system. A non-Lego solution was found consisting of an air cylinder and Festo pneumatic switches (Figure 13). The cylinder containing air at a pressure of 6 bar (600kPa) and Festo micropneumatic MZX switches could be controlled through the Lego interface. The Festo switches operate at 24 V DC and thus a set of relays and a transformer were built to operate these switches (Figure 9).

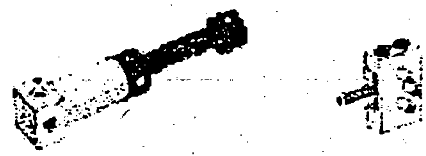


Figure 12 Lego Manual Pump and Switch (L-R).

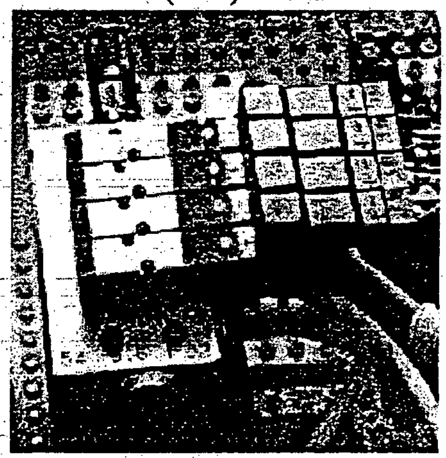


Figure 13 Festo MZX Switches.

3.4.3 Control of the Conveyor Sorting System

Lego Dacta components as well as the non-Lego components can be controlled with a personal computer. The CTRLab software package that is standard with the Lego interface can provide control for the standard Lego components. The software is designed for primary school children to operate and run simple applications. It is operated through the Logo language, which was unsuitable for the control of the Conveyor sorting system model to be constructed.

Lego developed standard subroutines for the control of the interface for various other languages. The language chosen for this project was Visual Basic 3.0.

The interface box (Figure 8) has 8 inputs and 8 outputs for the standard Lego components. The output ports provide up to 9 V DC. The pneumatic switches, however, require 24 V DC in order to operate correctly. A transformer and 4 relays (one for each of the 4 pneumatic switches) was built to step up the 9 V output to the required 24 V needed for the pneumatic switches.

The use of Visual Basic in this project as a programming language was made as it is a user friendly Windows object oriented language.

It was decided that the control of the CSS would be via a series of information windows and buttons.

3.5 Power Selection

The first design of the Conveyor sorting system utilised four Lego motors. Each motor would power one of the subsystems of the model. The subsystems are:

1. The Conveyor belt motor that drives the conveyor belt.
2. The "Feeder" which moves blocks from a magazine onto the conveyor belt.
3. The "Pushers" (there are two pushers) which push the blocks off the conveyor belt and into the their respective bins.

The power to the conveyor belt was supplied through a gear box (Figure 14).

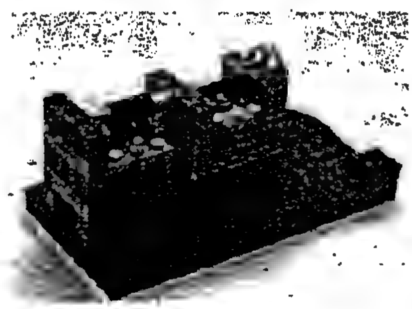


Figure 14 Lego Gear Box.

The initial design for both the feeder and pusher mechanisms was based on a crank arm and lever. The crank arm (Figure 15) was powered by the motor through a standard gear box (Figure 14) and the arm moved on a fixed horizontal plane. This, however, did not give the desired results as the speed of the lever arm was too slow. These mechanisms were replaced by pneumatic cylinders in the final design.

Figure 15 Lego Crank Arm.

3.6 Conveyor Design

The "conveyor belt links" join to form a belt (Figure 16). The belt was mounted on five 24 tooth gears (part No. 9966 Appendix 3), spaced equally along the length of the belt to provide support and to prevent the belt from sagging. The axles of the gears were mounted onto the side rails that run the length of the model.

The conveyer belt is rotated by a motor acting through a gearbox (gear ratio 3:5). Due to frictional forces in the mounting of the gear axes and the slip between the gears and the belt, the speed of the conveyor belt is not constant. This means that determining the position of the block on the conveyor belt at any time is difficult. To compensate for this, an angle sensor, that measures rotation, was placed on the first driven gear after the driving gear. The measurement of the angle sensor is translated into distance travelled along the conveyor belt. That is, when a block is fed onto the conveyor belt the distance to the required pusher is known and the distance elapsed can be measured. The pusher can then be activated when the block reaches its final position. Another advantage of this feedback loop was that should the conveyor belt stop and then restart the system would not be affected.

Figure 16 Lego Conveyor Belt.

3.7 Light Sensor Design

The Lego light sensor is an analogue sensor that measures light on a percentage scale. The sensor range of capabilities can be seen in Appendix 4.

The light sensor was originally placed over the conveyer belt looking down at the blocks as they passed below. It sampled continuous light readings which resulted in two problems:

First, light readings were given for the conveyer belt. This problem was compounded by the fact that the conveyer belt links have holes in them, which meant that the light sensor would see the black of the conveyer belt links followed by light shining through the holes in these links. This disturbance was difficult to filter out as the conveyor belt was not moving at a constant speed.

Second the light readings at the edge of the tile were different from that at the centre of the tile (at the edge of the tile, the light sensor received light both from the tile as well as from the conveyor belt). The problem was that the light signature of a tile was given as a bell curve. As the tile moved under the sensor, the leading edge would give a light reading, then a light reading would be taken at the centre of the tile followed by the trailing edge.

In order to solve both these problems it was decided to incorporate the light sensor into the feeder mechanism. This would have the advantage of eliminating background light sources and having the tile stationary during colour detection.

The light sensor was located in as dark a space as possible in order to eliminate outside light, rendering the light signal more useable. However, a small amount of stray light provided enough light to render the signal unusable. This was compensated for by taking a light reading before the tile was in place and again while the tile was under the sensor. Subtracting the background reading from the tile reading gave an absolute value for the tile independent of the outside light conditions. Another measure taken to ensure the accuracy of the light reading, was to record ten light readings of both the tile and the back ground and to calculate the mean for each of these values and work with the mean readings rather than a single one.

3.8 Feeder Design

The initial design for the feeder (Figures 17 to 19) was based on rectangular flat 1x2x1 Lego block. The feeder consisted of a crank arm and a lever, the crank arm being rotated by a Lego motor through a gear box and the arm moving in a fixed horizontal plane. The arm in its rear position would have a block drop in front of the lever arm by gravity and in the arm's full forward position it would push the block forward and on to the conveyer belt. The blocks would be stacked one on top of the other above the arm and as the arm moved back the stack of tiles would drop and a tile would be placed in front of the arm.

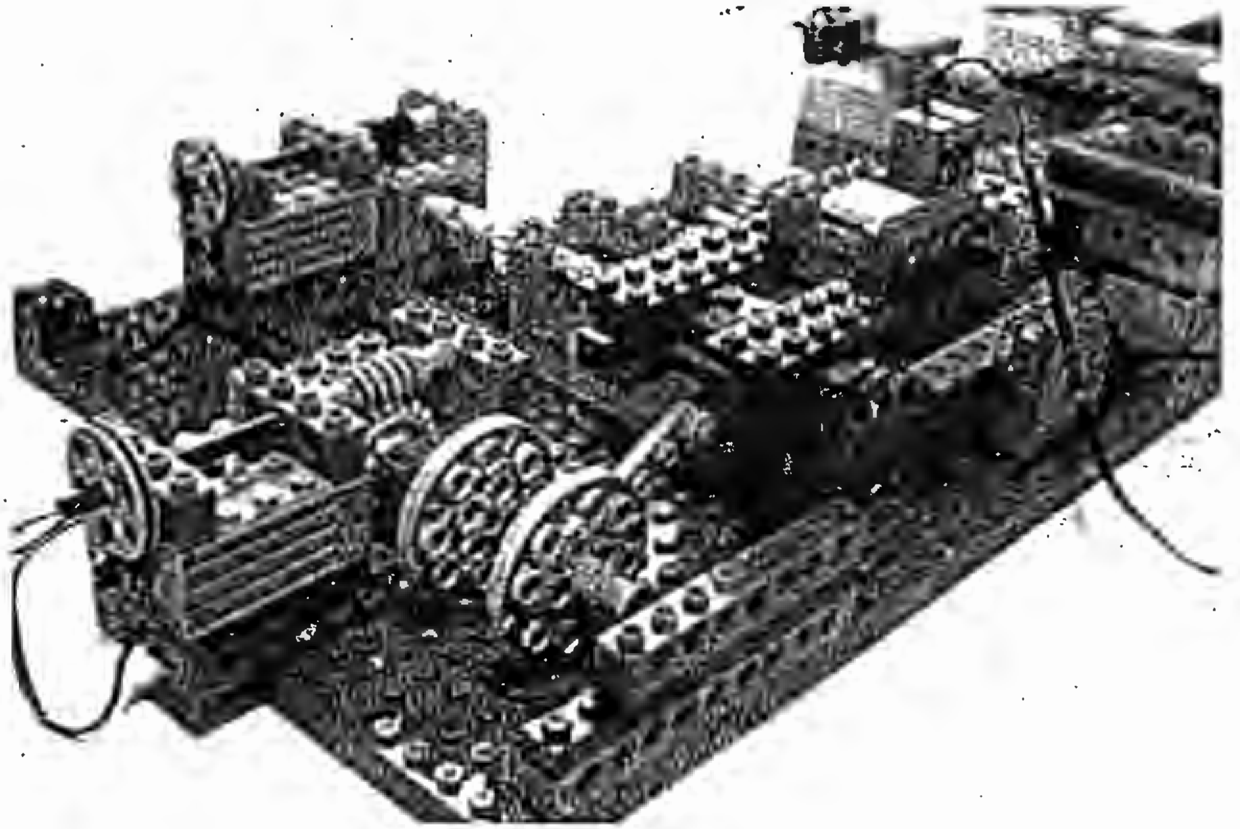


Figure 17 Feeder (Right View).

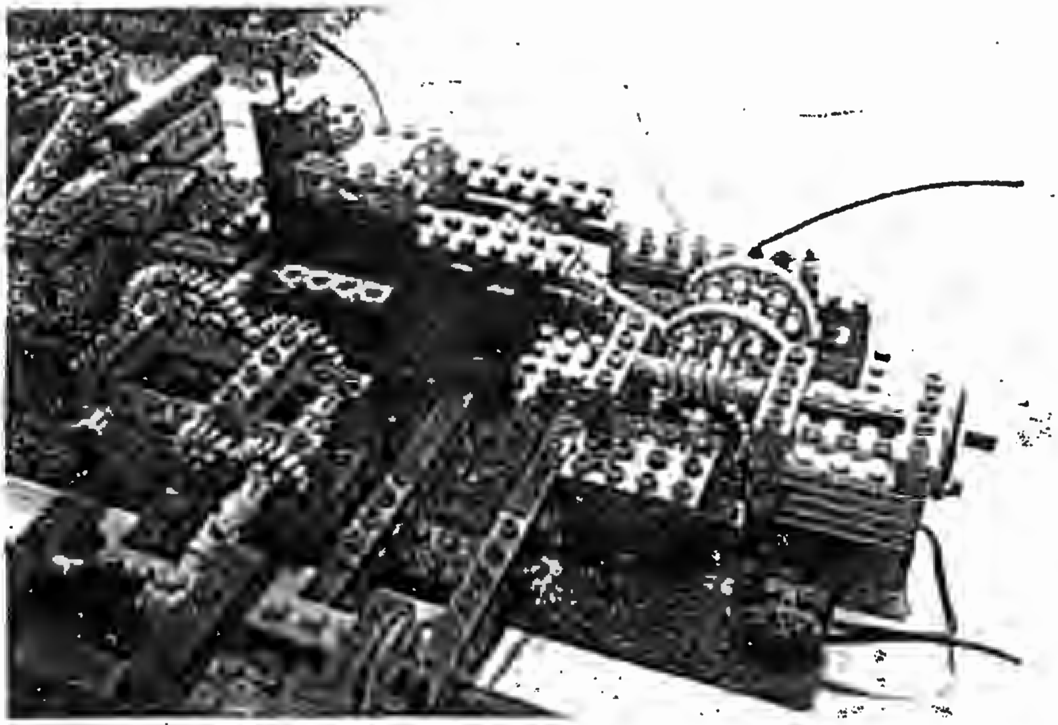


Figure 18 Feeder (Left View).

The feeder mechanism was built and tested. Two problems were found.

The first problem was that the blocks were not dropping in front of the arm. Secondly, the blocks dropping in front of the arm were often dropping at an angle, jamming the exit to the feeder.

Design and implementation of a suitable weight to sit on top of the stack of blocks in the magazine, forcing them to drop horizontally, into the correct position was attempted in order to overcome the problems. However, in testing, blocks fed correctly but the jamming problem reappeared. This it appeared was as a result of the arm moving slowly back and forward causing the block to drop at an angle as the arm moved backwards and remaining at an angle as the arm moved forward causing a jam.

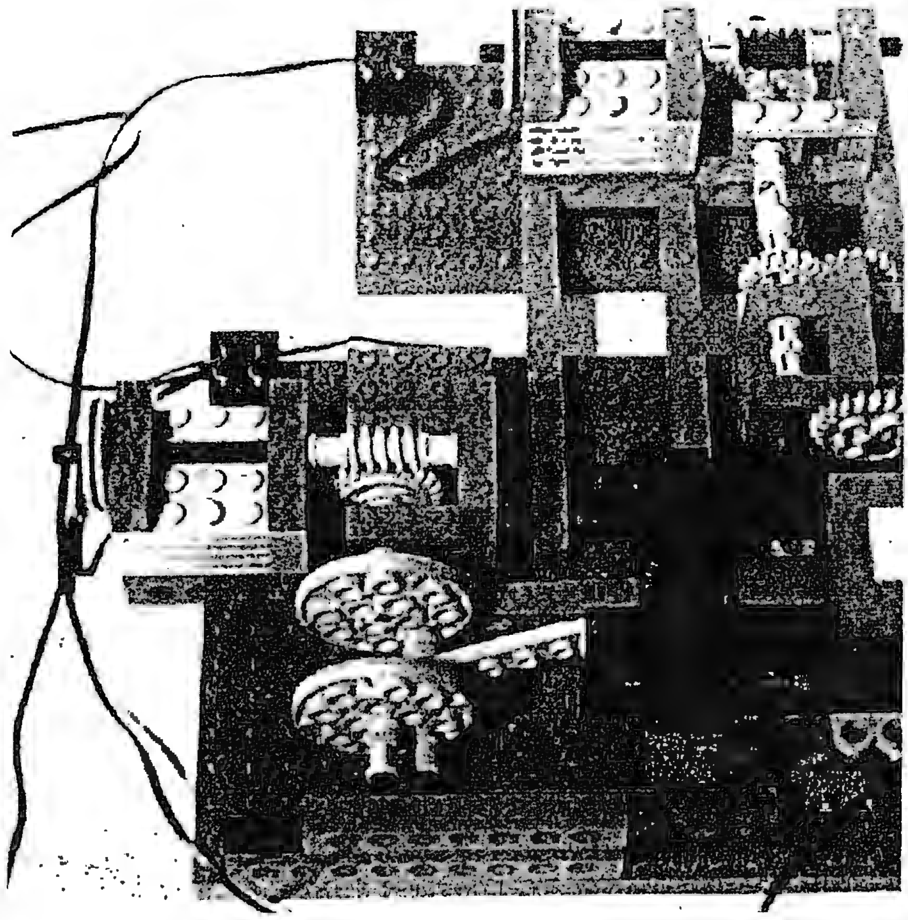


Figure 19 Feeder (Top View).

The best solution for this was to increase both the feeder speed and force. This could be done by adjusting the gear ratio of the gear box. However increasing the force caused a decrease in the speed resulting in the jamming problem.

Various modifications were made to the feeder including:

- Redesign of the feeder arm.
- Addition of angle sensors to the feeder arm to detect jamming and upon detection would retract the feeder arm and a dislodging forward movement to undo the jam.
- Change the 1x2x1 Lego block to a square 2x2x1 flat Lego block.

These changes did not solve the initial problems. It was finally decided that another approach to the feeder would have to be found. The only other method of exerting force within the Lego system was through the use of pneumatic cylinders (Figure 20). The disadvantage with these cylinders was that Lego only produced manual pneumatic switch gear for these cylinders.

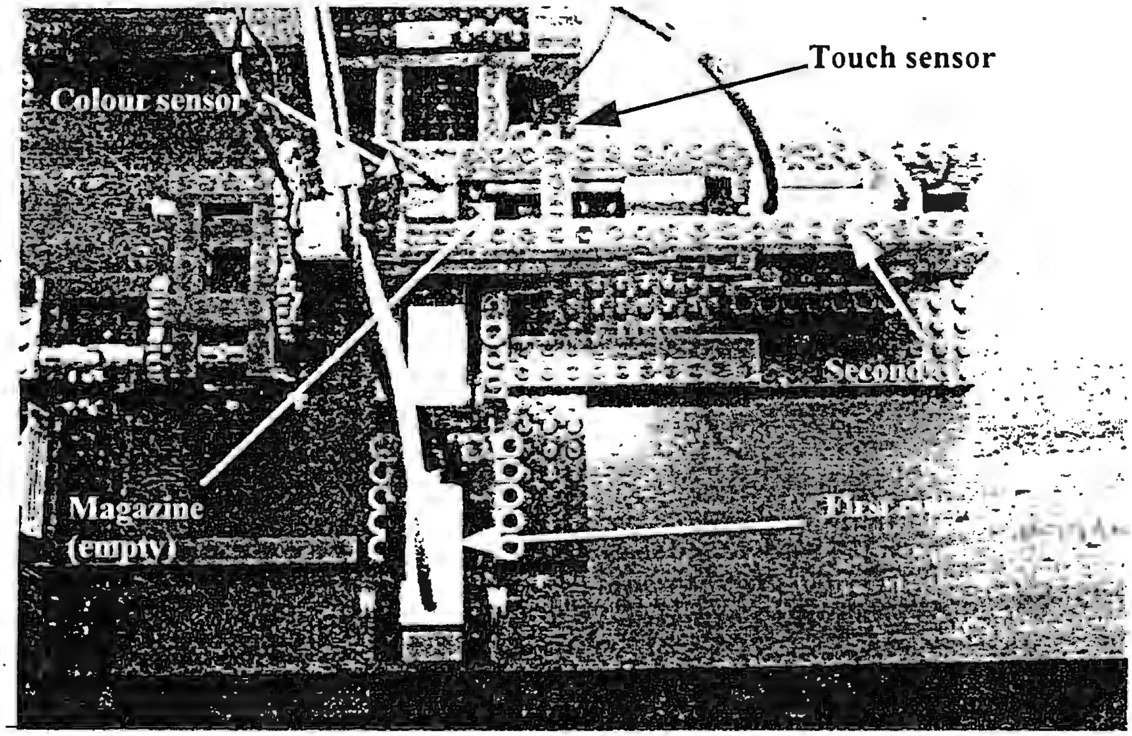


Figure 20 Final Feeder Configuration

After an investigation into commercially available electro-pneumatic switch gear was conducted, a suitable switch was found: Festo micro-pneumatic MZX 5/3 way valves (Figure 21). These valves operate at a pressure of 2 bar which is a suitable pressure for the Lego cylinders to operate optimally. A transformer and four relays were built (Figure 22) as the operating voltage of the valves was 24V DC. The valves were to be connected directly to the Lego interface, however the interface B only outputs 9V DC.

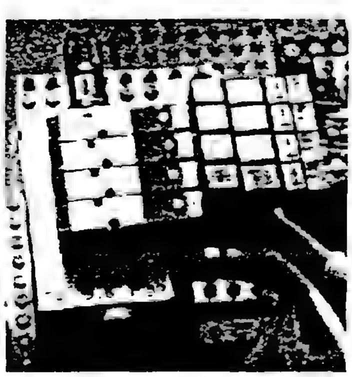


Figure 21 Festo MZX Switches

Through the relays and transformer the valves were able to be operated by the Lego interface B. A schematic of the relay circuit diagram can be seen in appendix 5.

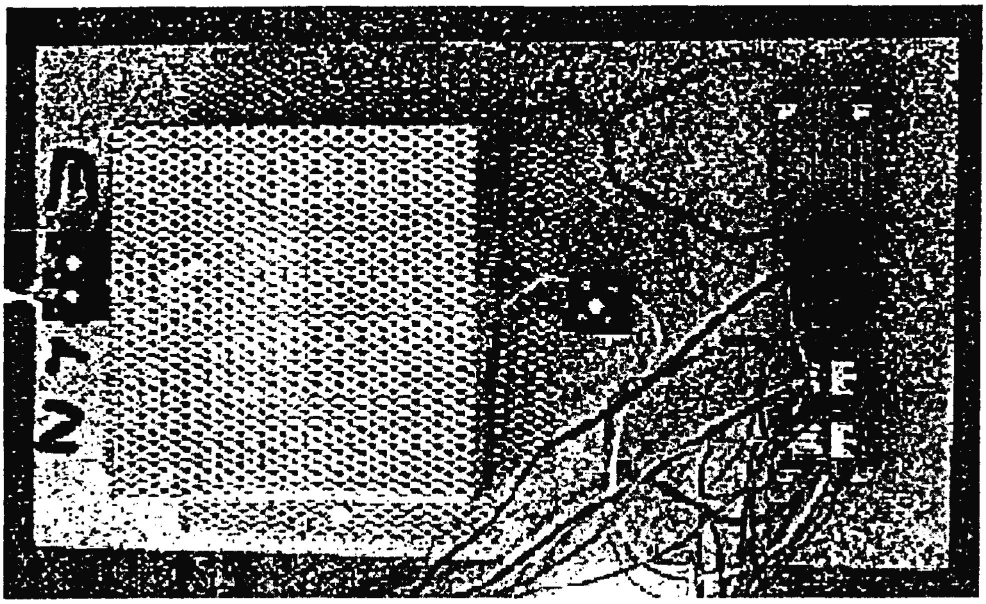


Figure 22 Transformer and Relays.

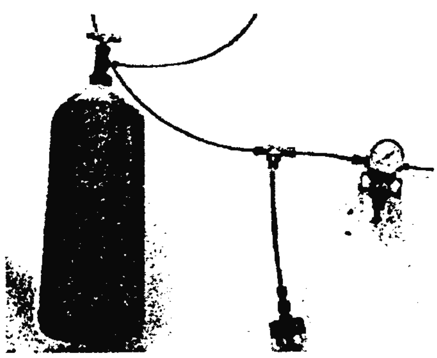


Figure 23 Air cylinder, Pressure Gauge and Pressure Reducer

Pressure to the pneumatic cylinders was supplied via an air cylinder charged to 6 bar and through a pressure reducer to 2 bar.

The design of the Feeder utilises two pneumatic cylinders and flat Lego 2x2x1blocks. These blocks were used as the design requires that the block be moved in two directions (both perpendicular and parallel to the conveyor belt) and a square block would pose less problems than the rectangular, 1x2x1 block.

The flat Lego 2x2x1 blocks were stacked in a magazine one on top of the other. The two pneumatic cylinders are arranged at right angles to each other, the first cylinder is to the right of the magazine (Figure 20) perpendicular to the conveyor belt and its function is to feed the block from the magazine to a point under the light sensor (see Figure 24 for the Feeder Logic Diagram). This action activates the push sensor which is located on the left hand side of the light sensor. Activation of the touch sensor shows that there is a block present under the light sensor and therefore a light reading must be taken. Should the magazine be empty and no block be fed to this position, the touch sensor will not activate the light sensor and the cylinders will retract to their original position. The second cylinder is situated behind the light sensor parallel to the conveyor belt. Once a reading has been taken, the second cylinder is activated and it pushes the block out from under the light sensor and on to the conveyor belt. The two cylinders then retract to their start position and the pile of blocks fall into place ready to begin the process again.

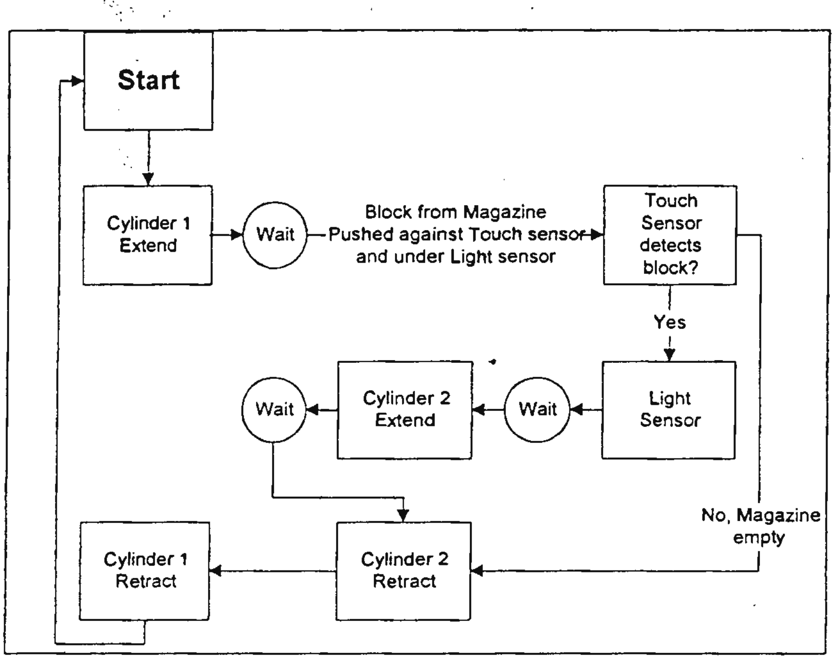


Figure 24 Feeder Logic Diagram.

3.9 Pusher Design

The pushers (there are two, one blue and one white) were originally based on the same rotary arm design as the feeder but the pusher arm in this case was a bar four wide (Figure 25 and 26) (Catalogue No. 9824, see Appendix 3).

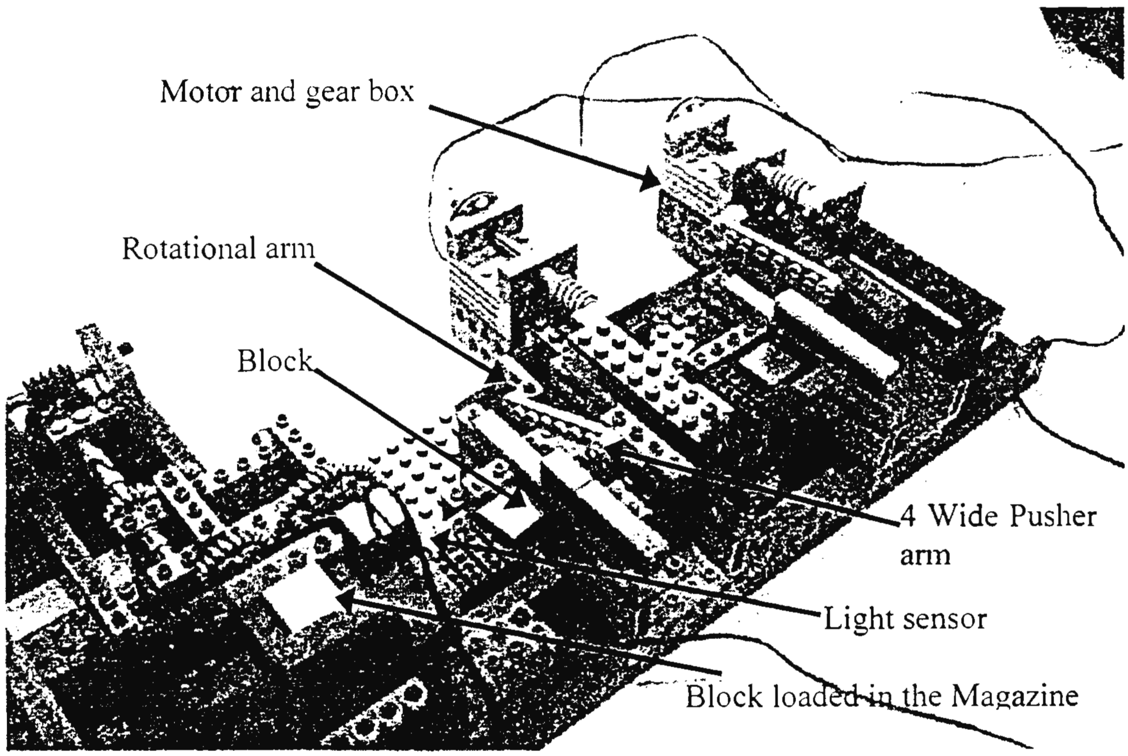


Figure 25 Pusher Mechanism (Top View).

The problems with this design were similar to those of the feeder. The slow speed caused tiles on the conveyer belt to move past the pusher arm before it could push the block off the conveyer belt. Increasing the speed of the pusher arm proved equally unsuccessful as the arm could not be stopped in the retracted position and blocked the conveyor belt. Slowing down of the conveyer belt caused the whole system to be extremely slow.

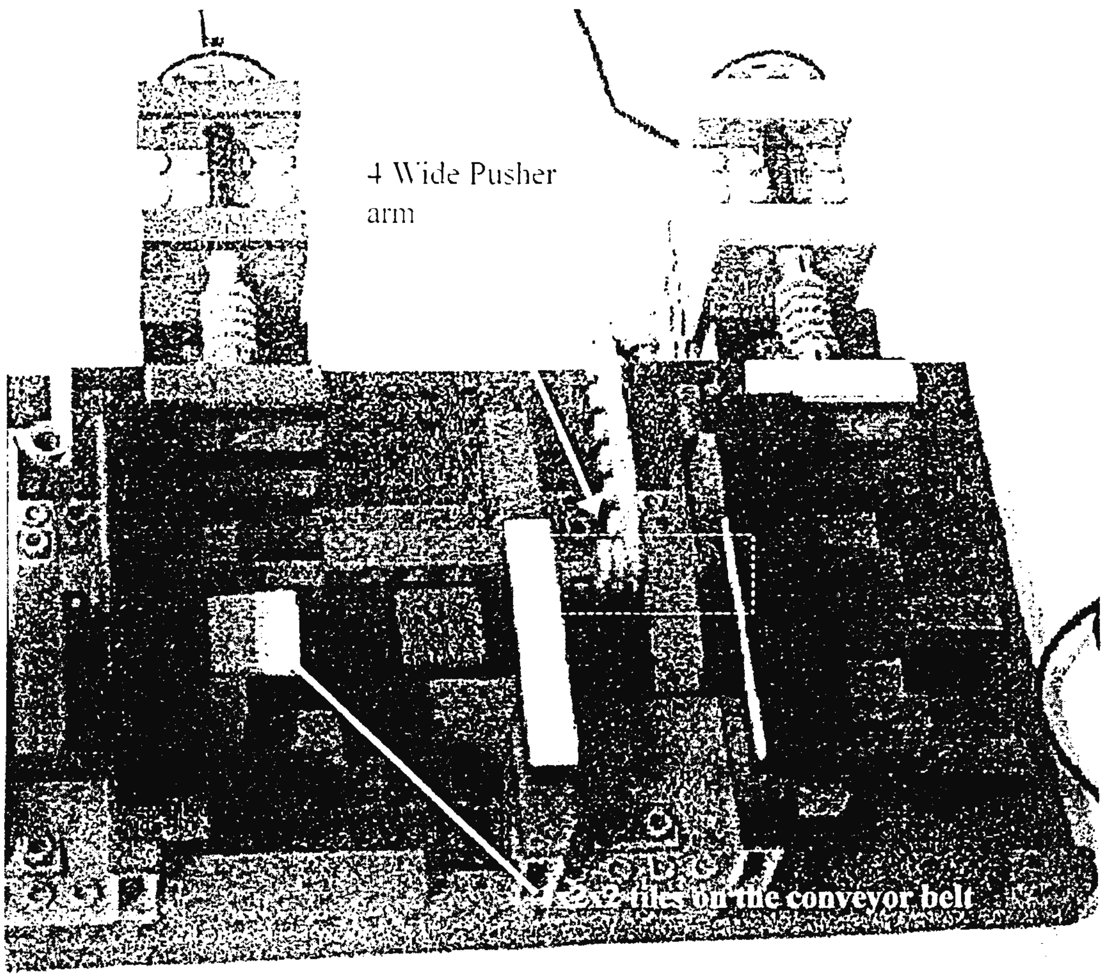


Figure 26 Pusher Mechanism (Side View).

As with the feeder mechanism, pneumatic cylinders were used to overcome this design problem.

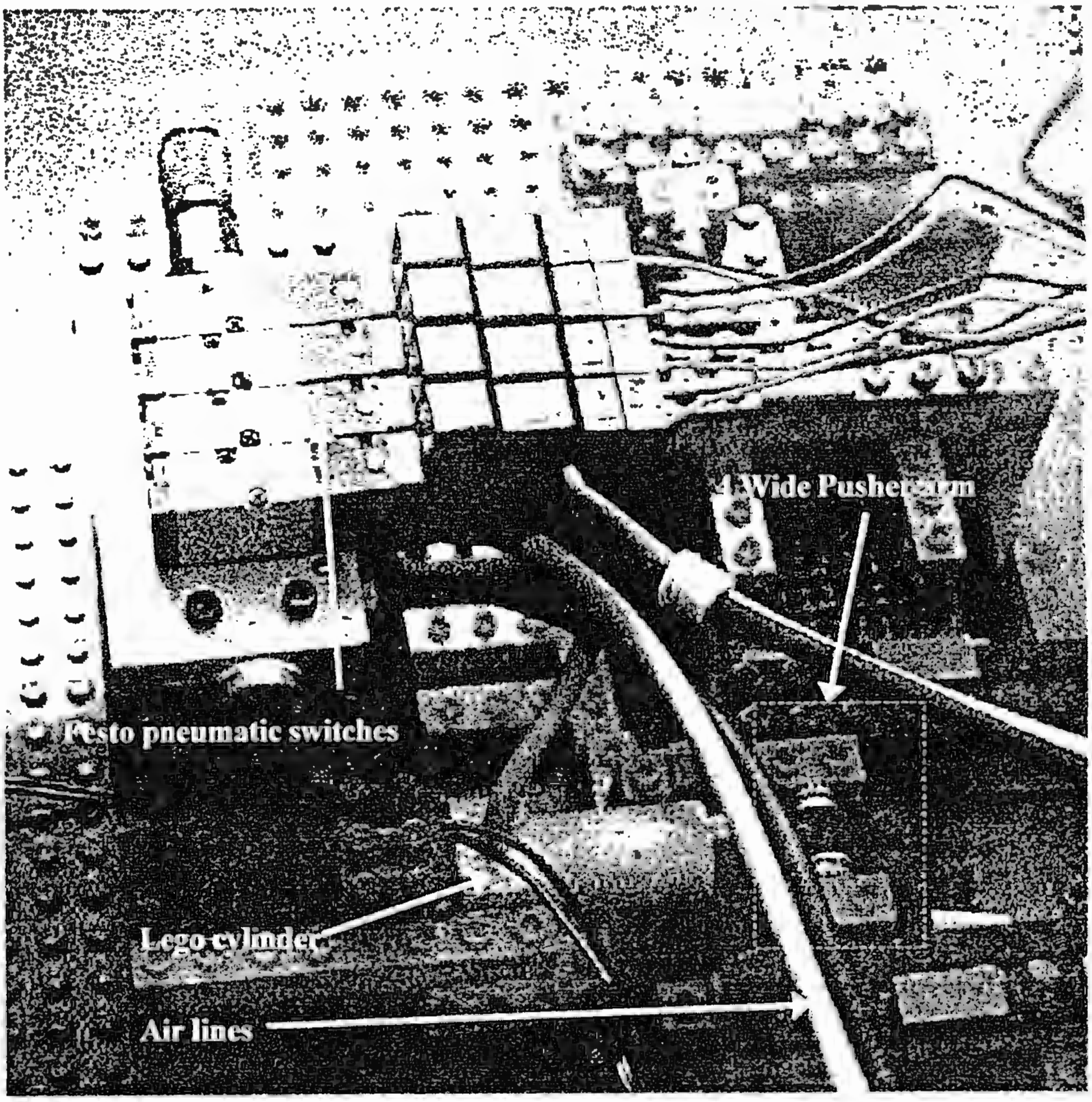


Figure 27 Final Pusher Configuration (Top View).

The pusher arm design remained a four wide block on a slider to constrain the arm in the horizontal plain (Figure 25 and 26). The rapid forward and return speed allowed for the conveyer belt speed to be increased. With this design. the arm could only be in either the retracted or extended positions. The final design for the pusher can be seen in Figure 27. The Pneumatic schematic diagram is in Appendix 6.

3.10 Control

In the Conveyor Sorting System, the functions are controlled by the computer through software and the Lego interface.

3.10.1 Inputs and Outputs

The CSS has 5 outputs and 3 inputs which are controlled by the computer software.

3.10.1.1 Outputs

The outputs to control are the motor and the four pneumatic cylinders. The motor speed is controlled by varying the output voltage and the cylinders are controlled by a voltage pulse. This is all controlled by the software and the Lego subroutines which control the interface box. The interface box is connected to the serial port of the computer. The motor is directly connected to the interface box. The pneumatic switches are connected through the relays to the interface.

The connections to the interface box are as follows:

Output A	Conveyor belt motor
Output B	Relay 3
Output C	Relay 4
Output D	Relay 1
Output E	Relay 2

3.10.1.2 Inputs

The inputs to the Conveyor sorting system are received from the light, touch and angle sensors. The light and angle sensors are analogue and the touch is a digital sensor. The interface box has eight input ports, four digital (yellow) and four analogue (blue).

Input 1 (Yellow)	Touch sensor
Input 5 (Blue)	Light sensor
Input 7 (Blue)	Angle sensor

3.10.2 Control Strategy

The Conveyor sorting system was developed to be a concept model for the further development of Lego as a modelling medium. Two strategies for the control of the Conveyor sorting system were developed; manual and automatic.

3.10.2.1 Manual control

The manual control allows the user to manually control as well as adjust the following:

- Switch the conveyor belt on and off.
- Manually operate the pushers.
- Activate the feeder mechanism completely manually or set an adjustable feed rate.

An emergency stop is provided should any part of the Conveyor sorting system malfunction or jam.

3.10.2.2 Automatic Control

In the automatic mode, the software controls the pushers. The feed rate can be activated manually or at a specific rate.

Chapter 4. The Computer Programme

4.1 Introduction

The control of the conveyor sorting system model is through software written in Visual Basic. Visual Basic was used as it provided easy object oriented windows to be built into the software for ease of use.

The Program controlling the model is CONVEYOR.EXE, a list of the code appears in Appendix 7.

When the programme CONVEYOR.EXE is run, the following windows are seen:

4.1.1 Information Window

This is a standard windows information dialogue box informing the user that the interface box is not correctly connected or is not switched on. The programme can not operate if the interface box is not properly connected. It is through this interface box that the program receives and sends information to the Conveyor Sorting System.

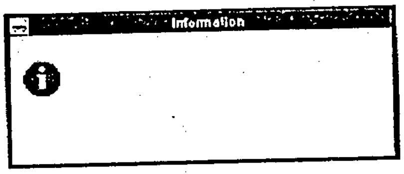


Figure 28 Information Window.

Once the interface box is correctly connected and switched on, the programme must be restarted. An example of the information box is given by (Figure 28).

4.1.2 Main Window

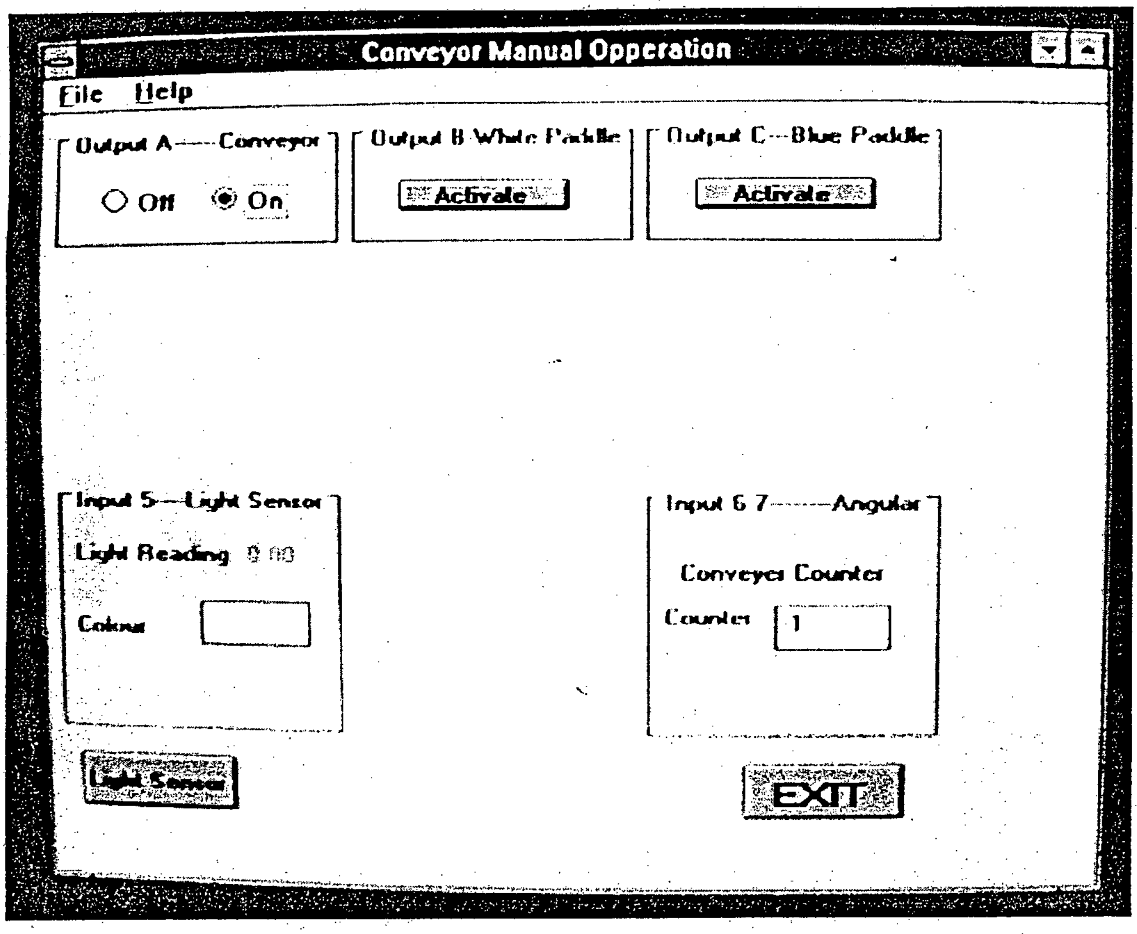


Figure 29 Main Window.

The main window (Figure 28) is the first window that appears if the interface box is correctly connected. This window allows the user to switch the conveyor belt on and off, activate either Blue or White Pusher, activate the Light sensor or EXIT the program. Activating the "Light Sensor" button from the first window (Figure 29), activates the Light sensor and this sensor takes 10 background light readings. Once this is completed, the Feeder Control button opens (Figure 30). Activating this button opens the Feeder Control window (Figure 30) with two options; manual and semi-automatic.

There is a toolbar with Exit and Help option. Exit will end the programme and stop all actions of the conveyor Sorting system. The Help option will call up either

HelpAbou.FRM (Information about the program version)or HelpRun.FRN (Information to help the user run the programme). A user manual appears in Appendix 8.

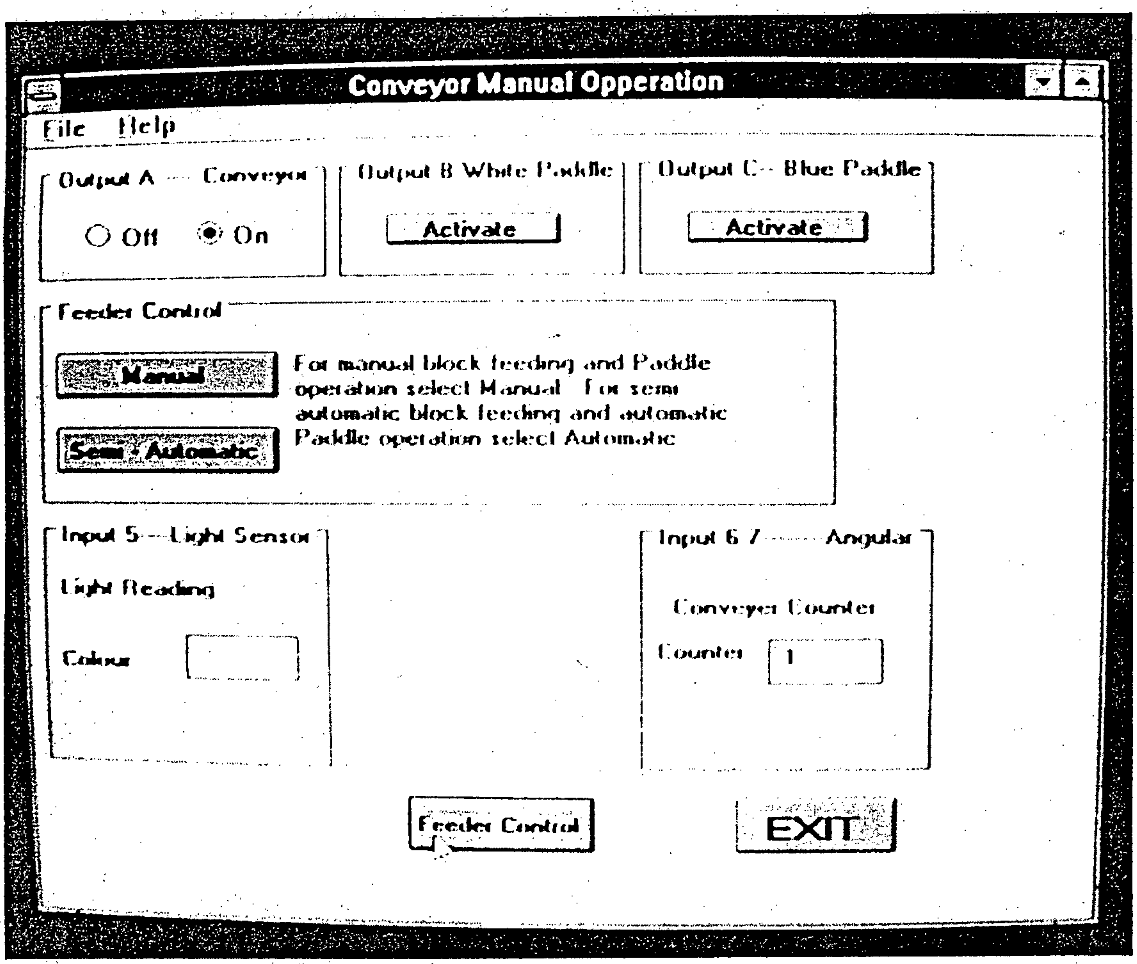


Figure 30 Feeder Control Window.

4.1.2.1 Manual Operation

Activating the "Manual" button from the Feeder Control window, (Figure 30) opens the "Output DE...Feeder" Window (Figure not shown) with only the Manual Feeder button. Activating the Manual Feeder button causes one block from the magazine to be released onto the conveyor belt after its colour has been determined. The block will continue along the length of the conveyor belt unless one of the Pusher buttons is manually activated.

4.1.2.2 Semi-Automatic Operation

Activating the SEMI-AUTOMATIC option from the Feeder Control window, (Figure 30) opens the Output DE...Feeder Window (Figure 31), with the Auto Activate button and slide bar as well as the STOP buttons. The Feeder Control "Output B White Paddle" and "Output C Blue Paddle" are disabled.

Activating the Auto Activate button starts the Model working. A Block will be released onto the conveyor belt after its colour has been determined at a rate set by the slider. Once the block reaches the appropriate Pusher, the software will activate that Pusher. This cycle will continue until the Magazine is empty.

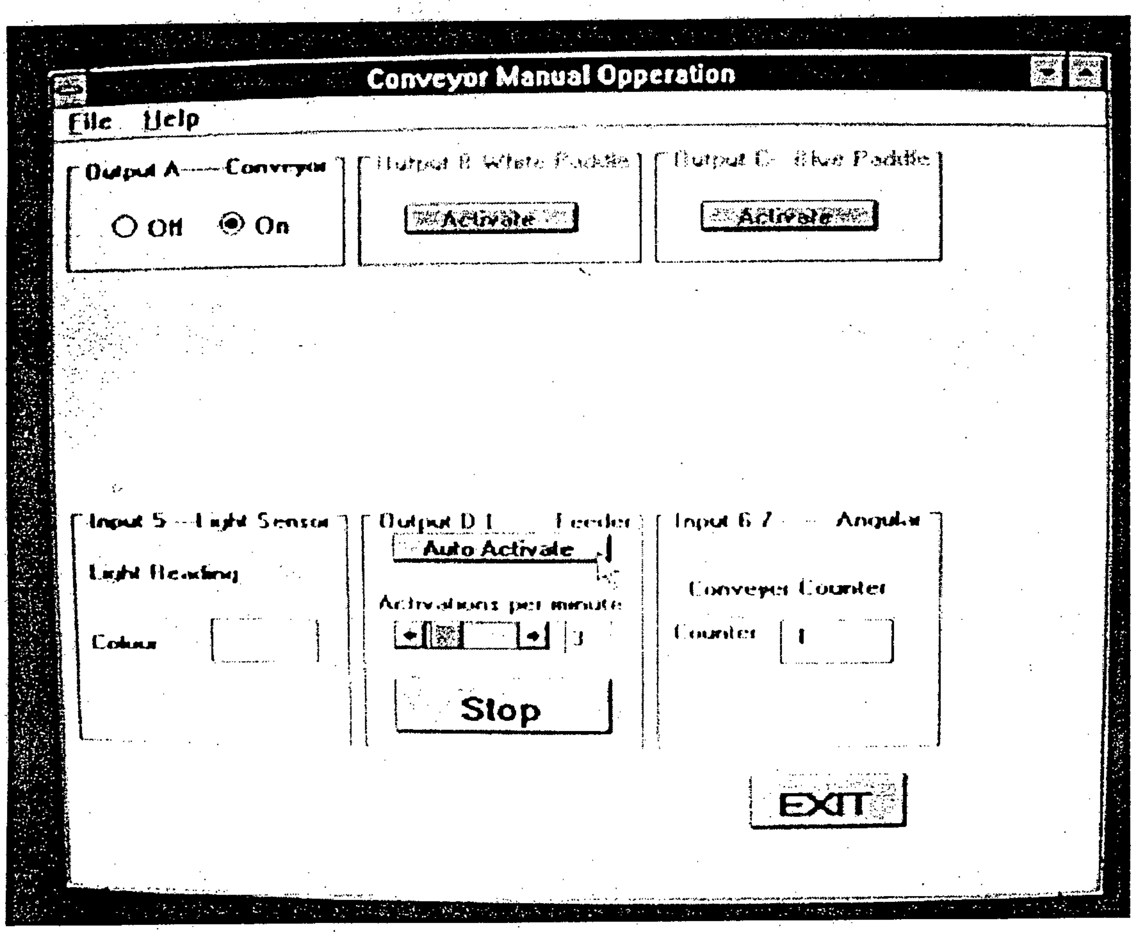


Figure 31 Output DE...Feeder Window.

Activating the STOP button at any time will stop the Models actions and take the user back to the MANUAL or SEMI AUTOMATIC choice in 4.1.2.

4.2 Control Algorithm

The control of the Conveyor sorting system is done through feedback from the various sensors. With the exception of the two pushers, all actuators are monitored or governed by sensors.

The angle sensor monitors the movement of the conveyor belt, allowing the software to determine when to activate either of the pushers.

The touch sensor detects the presence of a block under the light sensor and signals the software to begin light reading through the light sensor.

A flow diagram of the control programme is shown in (Figure 32).

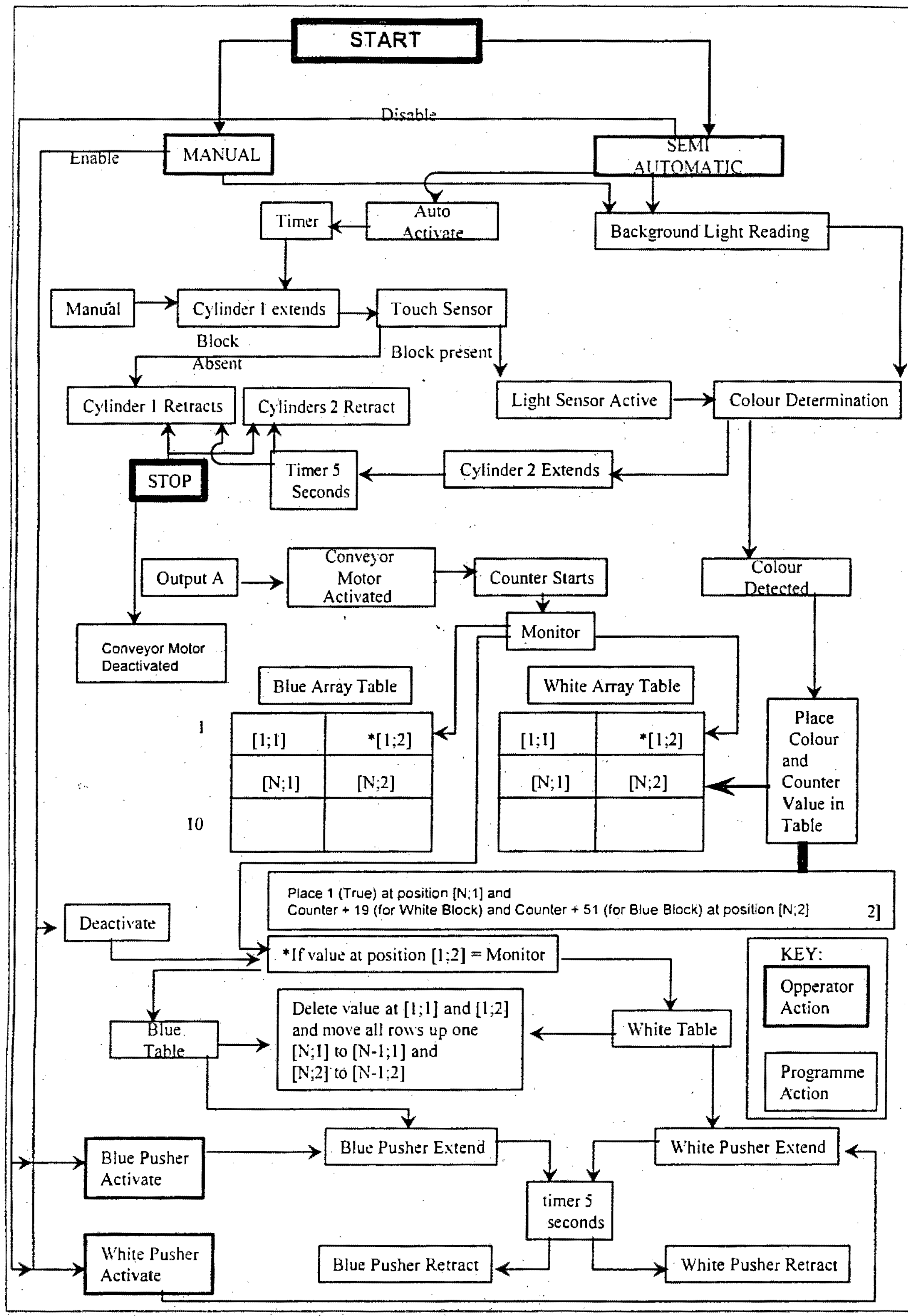


Figure 32 Flow Diagram of the Control Programme.

Chapter 5. Analysis of the model

5.1 Analysis

The conveyor sorting system was analysed in terms of the objectives, the viability of Lego as a simulation tool and the viability of Lego as an educational tool.

The model cannot be tested in terms of its components, but in terms of its overall function. The model achieved its objectives of sorting randomly stacked blue and white blocks into their respective colours. The process of sorting the tiles can be seen on the attached VHS cassette tape, (Addendum A). A brief description of the various components making up the conveyer sorting system and several demonstrations of the working model can be seen.

The software controlling the model was designed simply to control the model. The only parameter that can be adjusted is the feeder. The feeder can be set on automatic, that is, it feeds tiles onto the conveyer belt at a set rate. Alternatively the tiles can be released manually, that is, at the discretion of the operator. Other parameters, such as: Variable and Constant time operation, variable conveyor speed and additional coloured tiles to be sorted were not included in the software, as they were beyond the stated objectives of the study.

Chapter 6. Conclusions

6.1 The Model

The conveyor sorting system was tested a number of times, as can be viewed on the VHS video cassette (Addendum A) and all trials were successful. It can be concluded that the model achieved the objective of sorting two differently coloured, randomly placed blocks into their respective coloured bins.

6.2 Primary Objectives

The primary two objectives; the viability of Lego as a simulation tool and the viability of Lego as an educational tool, require a more analytical approach.

6.2.1 Lego as a Modelling Tool

The conveyor sorting system in its present configuration is of no use as a simulation tool. Possible additions to the current model would be:

- Increased number of operations (such as additional processes along the conveyor belt).
- Increased number of colours (allow the operator to choose the number of colours to be sorted).
- Addition of variable and fixed time operations allowing the operator to program the time each operation requires.

However the objective of the model was not to achieve this function, but rather to evaluate the potential of Lego as a simulation tool. In this respect, the model demonstrated that basic functions required for a complex model to exist. Therefore Lego is viable and has the potential to be a valuable modelling tool.

6.2.2 Lego as an Educational Tool

Lego, through this project had demonstrated to be an effective learning tool. From the self propelled car [6:Martin, Resnick, 1993] to the robots of MIT [7:Martin, 1992], from the undergraduate ASPECT programme [8: Sass, 1988] to this project Lego has proven to be a viable, valuable and useful educational tool.

In school, Lego has proven to be a very useful in the running of the "Introduction to Technology" [5:DeSousa,1994]. Although this programme is not within the formal school curriculum, it is advantageous to the continuing education of the students.

In undergraduate study Lego was found to be useful as an educational tool in teaching:

- Principles of engineering such as gears (gear boxes and differentials) levers (first, second and third class) and pulleys.
- In design, its usefulness in improving engineering communication and Lego's reusability, flexibility and low cost make it an ideal tool.
- In simulation where students could be given a Lego model, such as the conveyor sorting system and have to design the controlling software to operate the model optimally.

In the Postgraduate study, it could be used as a simulation medium (an upgraded Conveyor Sorting System) for both production lines or robotics, as was accomplished in the following projects:

Hrabar S.E, "Automation Of A Warehouse Packing And Inventory Control", this thesis developed a system that packs and unpacks a warehouse automatically, while keeping record of the stock.

Tollner M.E, "Modelling Of Mechanical Systems Using Computer Interactive Devices", this thesis developed a model of a fully automated factory packaging system.

Tollner M, "Designing And Modelling An Automated Packaging System", this thesis developed an improved materials handling system for a beverage cap production factory.

Kotze, J. "A Concept Model For A Multi-Fingered Prosthetic Hand", this thesis designed a computer controlled and operated prosthetic hand.

References

1 Dickens, C., "Hard Times", Collins, 1954.

2 Toffler, A., "The Third Wave", Collin's, 1980.

3 Bennet, W., "First lessons: A report on Elementary Education in the United States", US Dept. of Education, 1986.

4 Pappert, "Mindstorms", Basic Books Inc., 1980.

5 DeSousa K., Lego turns kids onto Engineering", Engineering News, Vol. 4 number 28, 22 July 1994, pp 100.

6 Martin F., Resnick M., "LEGO/Logo and Electronic Bricks: Creating a Scienceland for Children",, "Advanced Educational Technologies for Mathematics and Science", Eds. Ferguson D.L, Springer-Verlag, 1993.

7 Martin F.G., "The 6.270 Robot Builder's Guide for the 1992 MIT LEGO Robot Design Competition", Media Laboratory MIT 1992.

8 Sass, A., "Academic Support in Engineering at the University of Cape Town", Journal of Higher Education, Vol. 2, 1988.

9 UCT, "Faculty of Engineering Student Hand Book", University of Cape Town Press, Book 7, 1989.

10 Jawitz, J., "Report on the introduction to mechanics workshops pilot programme", Faculty of Engineering, UCT, Nov. 1992.

11 Rowland, J.R., "Engineering Models Still Teach and Sell", Machine Design, Dec. 11, 1986, pp 56-60.

12 CNN Science and Technology Week, "Computer Simulation Aids in Development of New Planes", An interview with Gulfstream Aerospace, aired on January 20, 1996.

13 Senge, P., "The fifth Discipline, The art and practice of the learning organisation.", Doubleday, 1990.

14 Chase R.B., Aquino N.J., "Production and Operations Management, A Life Cycle", Fifth Edition, Irwin, 1989.

15 Ackoff R., "Toward a Systems Theory of Organisation", Strategic Management Journal, Vol. 5, 1984.

16 Hrabar S.E, "Automation Of A Warehouse Packing And Inventory Control", Postgrduate Thesis, Dept. Mechanical Engineering, UCT, November 1996.

17 Tollner M.E, "Modelling Of Mechanical Systems Using Computer Interactive Devices", Undergraduate Thesis, Dept. Mechanical Engineering, UCT, Nov, 1995.

18 Tollner M, "Designing And Modelling An Automated Packaging System", Postgraduate Thesis, Dept. Mechanical Engineering, UCT, 1997.

19 Kotze, J. "A Concept Model For A Multi-Fingered Prosthetic Hand", Postgraduate Thesis, Dept. Mechanical Engineering, UCT, 1997.

Bibliography

20 Spriet J.A., "Computer Aided Modelling and Simulation", Academic Press, 1982.

21 Maisels H., Gnugnio G., "Simulation of Discrete Stochastic Systems", Science Research Associates Inc., 1972.

22 Linstone H.A., Simmonds C.W.H., "Future Research View Directions", Addison-Wesley, 1972.

23 Oren T.I., Zeigler B.P., Elzas M.S., "Simulation and Model Based Methodologies", Springer-Verlag, 1984.

24 Cesar J., Sitter M., "Lego and Logo; Time and Space", Eds. Johnson DC, Samways B, "Informationatics and changes in Learning", IFIP Transactions A, North-Holland, 1993.

25 "British Try Computer Aided Lego", Engineering News Record, Vol. 214, pp 21, May 1985.

26 Jaklic A., Volk T., Leonardi M., "Design and Performance Analysis of a Simple Mobile Robot", Elektrotehniski Vestnik, Vol. 60, Part 4, 1993, pp 233-239.

27 Tiplady P., "Modelling and Simulation (STELLA)", New Scientist, v115, pp 76-77, 24 September 1987.

28 Chaharabaghi K., "Using Simulation to Solve Design and Operational Problems", International Journal of Operations Management, pp 38-31, January 1990.

29 Almeleh M.R., "Computer Simulation for Manufacturing Systems", BSc Thesis, UCT Department of Mechanical Engineering, Cape Town, Nov. 1992.

30 Stix G., "Plane Geometry: Boeing uses CAD to design 130,000 parts for its new 777", Scientific American, v 264, pp 110-111, March 1991.

31 Lego Dacta, "Directions for Use", from Lego box set 9701, Made by LEGO Systems A/S, DK-7190 Billund, Denmark.

32 Hurrion R.D., "Simulation: Applications in Manufacturing", Interfaces, v19, p122-123, January-February 1989.

33 "Modelling Takes the Guess Work out of Industrial Processes", Research & Development, V32, pp78-82, April 1990.

34 Greenfeild C., Porter B., "Modular Visualisation of Organisational Structures:- An implementation Model for Re-engineering Programmes", 13th Annual Conference of the Irish Committee Re-engineering for World Class Manufacturing, University of Limerick, 4-6th September 1996m

35 Ackoff R.L., "Ackoff's Fables", John Wiely & Son, 1991.

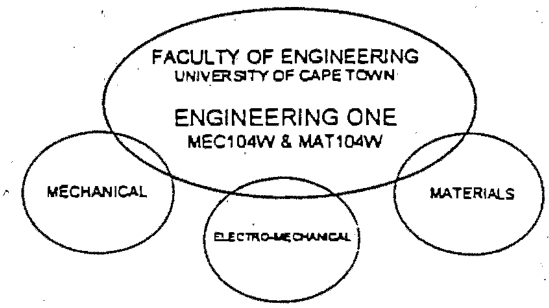
36 Upton D.M., McAfee A., "The Virtual Factory", Harvard Business Review, vol. 74, No 4, July-August 1996, pp. 123-133.

37 Wali S., "Simulation Software Products for Analysing Manufacturing Systems", Industrial Engineering, V18, No. 7, pp 98-103, July 1986.

List of Appendices

Appendix 1 Design I Course Problem Sheet

(Total of 15 pages)



Gear Project

Day 1 - Classwork and Homework Task

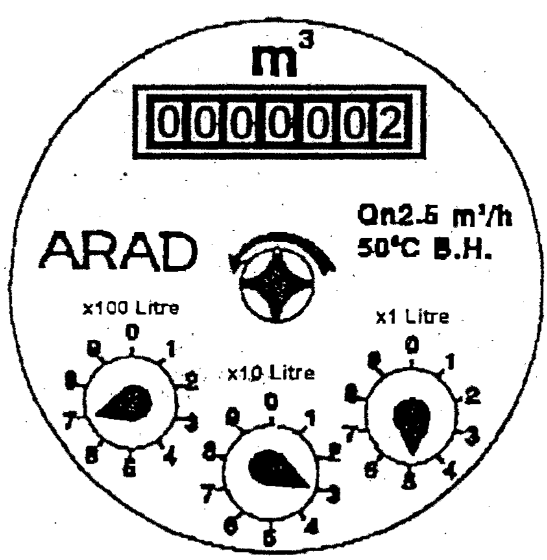
Read the handout entitled **The Tooth about Gears** and prepare yourself for a written test on **Day 2**.

Complete the following task to be handed in on **Day 2**. Be sure to include your name on the top right hand corner of the page that contains your answers. Combine the information given on pages 4 & 5 (Gearing Up / Gearing Down) and page 6 (Idler Gearing) to produce a **DRIVE CHAIN** that can be used to drive the three pointers on the water flow meter shown below (Can you find the error in the schematic of the water flow meter ?). Note that the pointers rotate in the same direction and indicate the volume flow shown above each dial. The present reading is 2 m^3 and 735 litres.

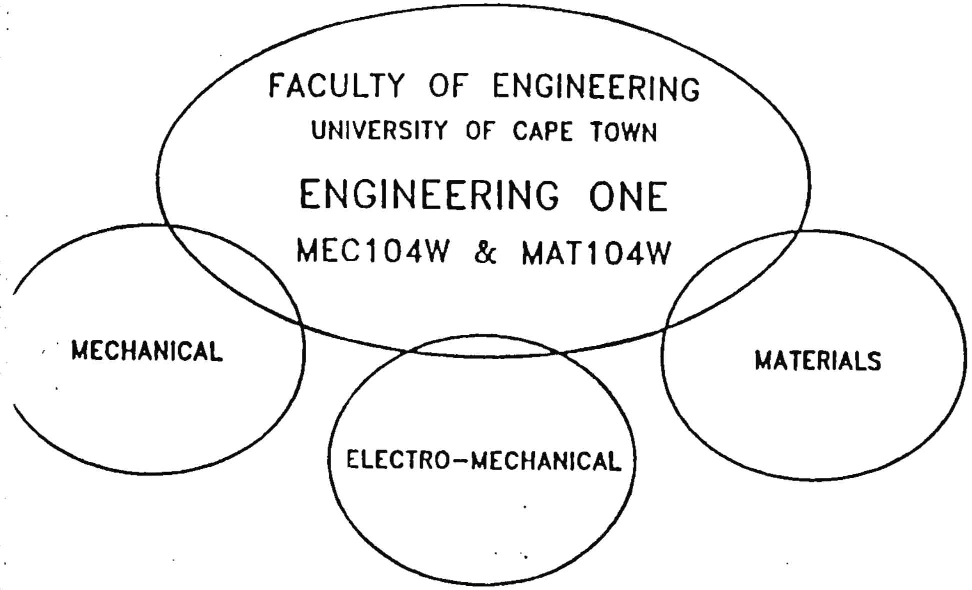
Day 2 - Classwork Task to be completed in groups of two

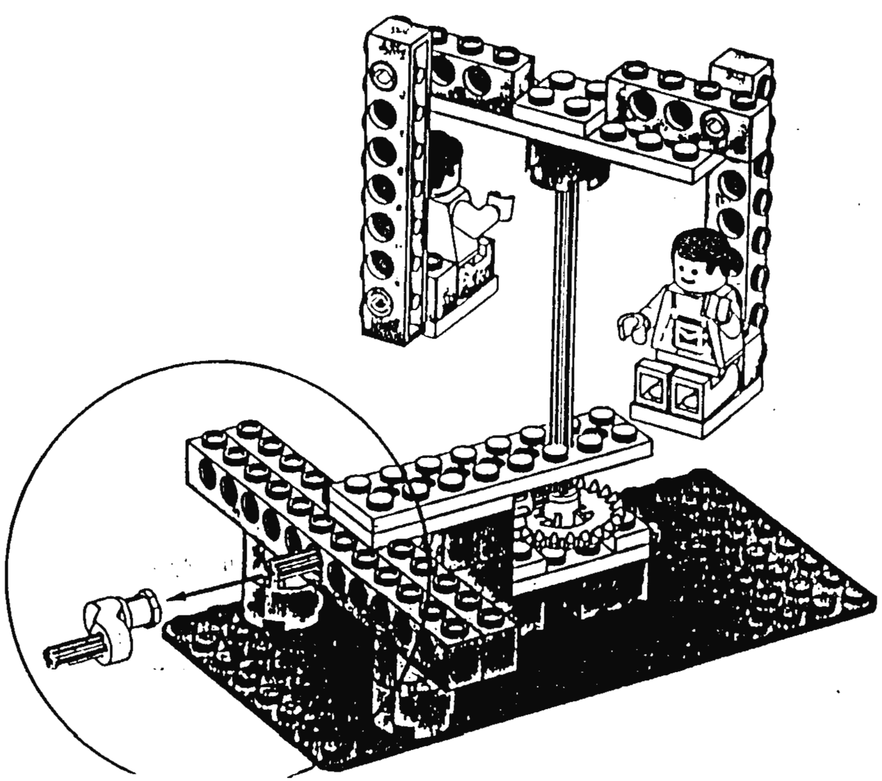
Demonstrate that you have done the above by completing a written test. (No notes allowed during this test)

- Build the LEGO model shown on page 1 of the handout **The Tooth about Gears**.
- Predict if the top will turn faster or slower than the handle.
- What sort of gearing arrangement is this ?
- Determine the gear ratio.
- Modify the model to improve the situation.
- What is the gear ratio now ?



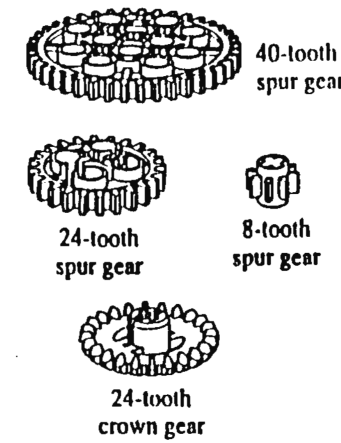
3 POINTER WATER METER





The Tooth about Gears

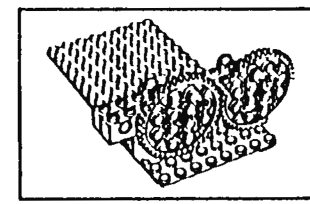
A gear is a toothed wheel, one of the simple machines. Several examples from the LEGO DACTA® Gear Set are shown on the right. Gears can be used to transfer force, increase or reduce speed, and change the direction of rotary motion.



Gears are all around us. The photographs on the back of this booklet show examples of devices with gears.

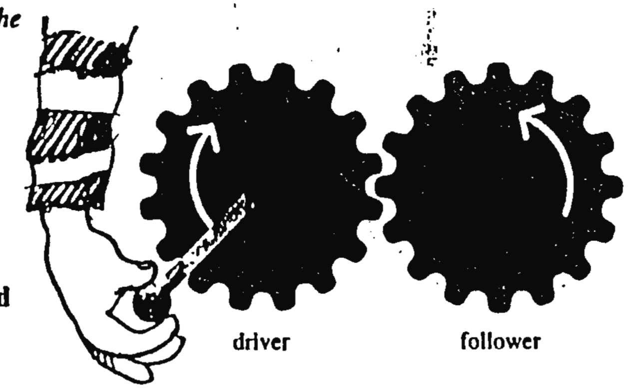
Familiarize yourself with gears by working through activities on pages 3–7. Photocopy the drawings on pages 16–17 for easy reference. You may wish to provide these activities to your students as well.

Direction of Rotation



◆ Build the base and model 1 shown on page 16. Insert connector pegs in the gears to act as handles. Predict what happens when you turn one of the handles.

■ *The gear you turn (called the driver) causes the other gear (called the follower) to turn also. The driver and the follower turn in opposite directions.*

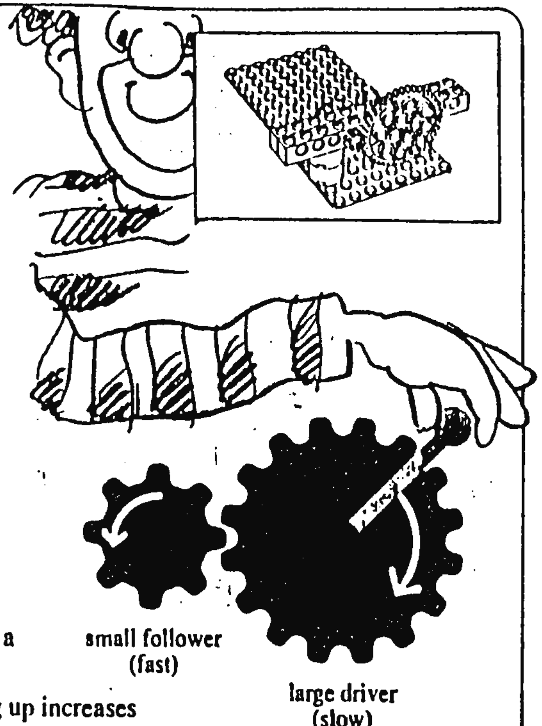


Main Idea:

Two gears which are meshed turn in opposite directions.

The Tooth about Gears

Gearing Up



◆ Build model 2 shown on page 16, using one 40-tooth gear and one 8-tooth gear. Predict what happens when you turn the handle on the 40-tooth gear.

■ *The small 8-tooth follower turns faster than the 40-tooth driver. The driver and the follower also turn in opposite directions.*

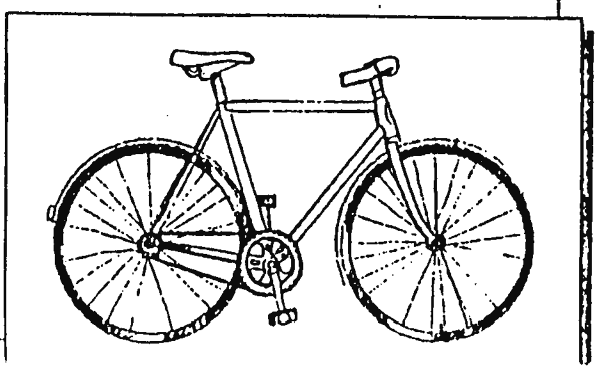
Main Idea:

A large driver gear makes a small follower gear turn faster.

Additional Information:

A large driver gear turning once can make a small follower gear turn many times. This arrangement is called **gearing up**. Gearing up increases the speed of rotation, but decreases the force. To go fast on a 10-speed bike, you shift into 10th gear. In this mode, you turn a large gear with the pedals which drives a small gear attached to the rear wheel. One turn of the pedals produces about four turns of the rear bicycle wheel in 10th gear.

For the LEGO® model you built, one turn of the 40-tooth driver produces **five** turns of the 8-tooth follower. This ratio of 1:5 (or 1/5) is called the **gearing up ratio**. (8/40 = 1/5).

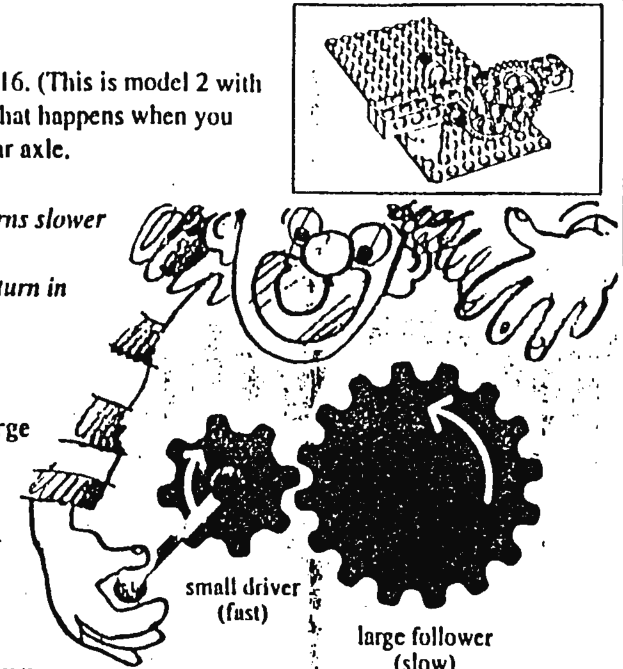


Extension:

Build a model using other gears of different sizes for gearing up.

The Tooth about Gears

Gearing Down



◆ Build model 3 shown on page 16. (This is model 2 with a small handle added.) Predict what happens when you turn the handle on the 8-tooth gear axle.

■ *The large 40-tooth follower turns slower than the 8-tooth driver. The driver and the follower also turn in opposite directions.*

Main Idea:

A small driver gear makes a large follower gear turn slower.

Additional Information:

A small driver gear has to turn several times to make a large follower gear turn once. This arrangement is called **gearing down**. Gearing down decreases the speed of rotation but increases the force. When you shift a car into first gear to climb a hill, you are gearing down.

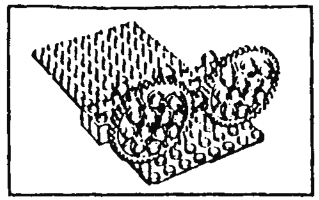
For this model, **five** turns of the 8-tooth driver produce one turn of the 40-tooth follower. This ratio of 5:1 (or 5/1) is called the **gearing down ratio**. (40/8 = 5/1).

Extension:

Build a model using other gears of different sizes for gearing down.

he Tooth about Gears

dler Gearing



• Build model 4, as shown on page 17. Predict what appens when you turn one of the gear handles.

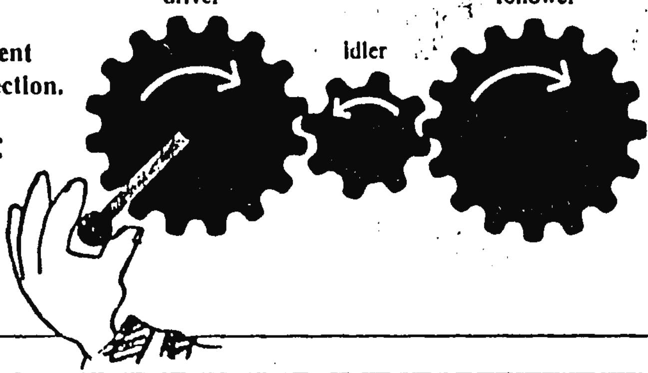
■ *The 40-tooth driver and the 40-tooth follower both urn in the same direction. The gear in the middle the idler) rotates in the opposite direction.*

Main Idea:

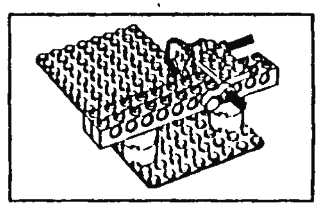
An idler gear makes adjacent gears turn in the same direction.

Additional Information:

A large driver will make a small follower turn faster, regardless of the idler size.



At An Angle



◆ Build model 5 shown on page 17, using the crown gear and an 8-tooth gear. Predict what happens when you turn the handle.

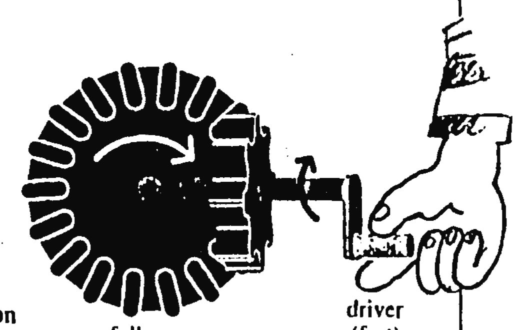
■ *The rotary motion goes through a 90-degree angle. The 24-tooth crown gear follower turns slower than the 8-tooth driver.*

Main Idea:

The crown gear can change the rotary motion through a 90-degree angle.

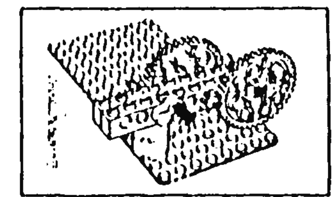
Additional Information:

This model has a gearing down ratio of 3:1. (24/8 = 3/1). See also the top two photographs on the back of this booklet for additional examples.



The Tooth about Gears

Compound Gearing



◆ Build model 6 shown on page 17. Predict what happens when you turn the handle on the 8-tooth gear.

■ *The first 40-tooth gear turns slowly. The second 40-tooth gear turns even more slowly.*

Main Idea:

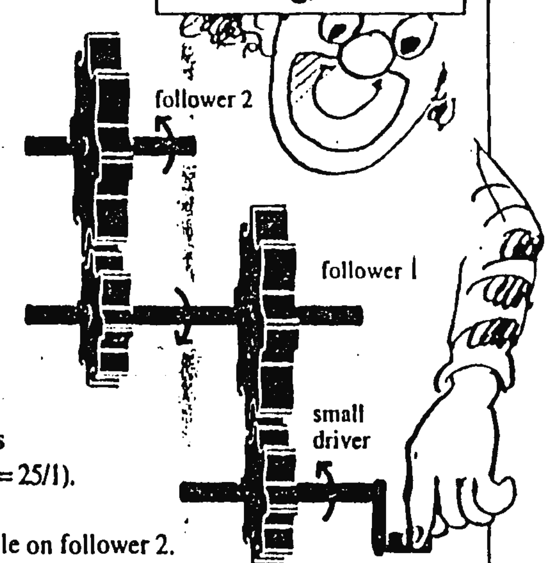
Gears of different sizes on the same axle can be connected to other gears to build more extensive gearing down (and gearing up) arrangements.

Additional Information:

Two separate 5:1 gearing down arrangements are connected to each other by the axle passing through the first 40-tooth gear and the second 8-tooth gear. This connection increases the gearing down ratio to 25:1. (5/1 times 5/1 = 25/1).

Extension:

Predict what happens when you turn the handle on follower 2. *The final 8-tooth gear rotates very fast, because of the 1:25 gearing up ratio. (1/5 times 1/5 = 1/25).*

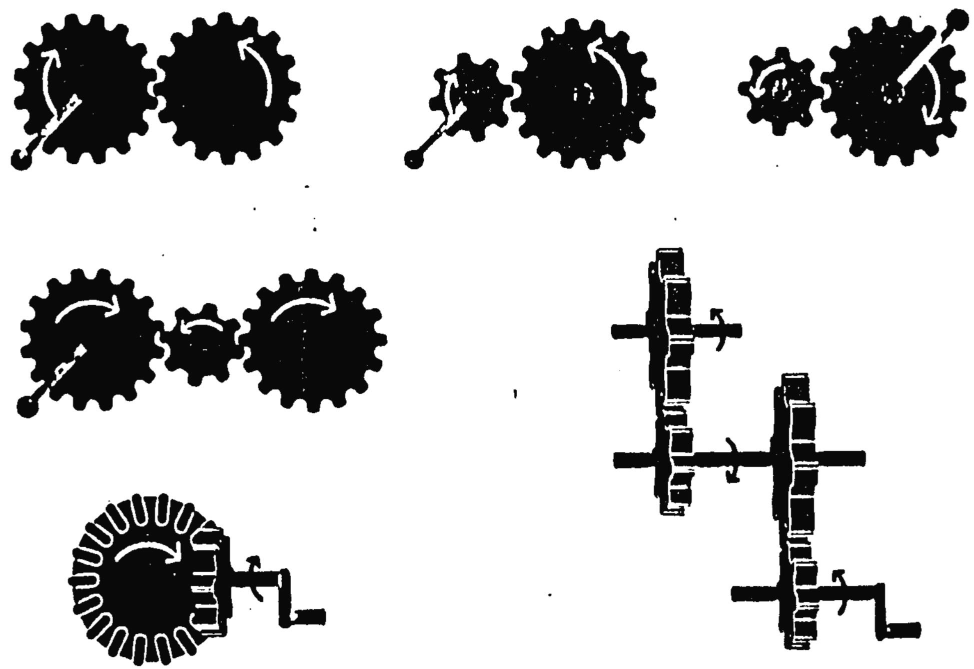


Gear Summary

■ Two meshed gears turn in opposite directions.
■ A large gear makes a small gear turn faster (gearing up); a small gear makes a large gear turn slower (gearing down).
■ An idler gear makes gears meshing with it turn in the same direction.
■ A crown gear helps change motion through 90 degrees.
■ Gears of different sizes on the same axle can be used to build more extensive gearing up or gearing down arrangements.

Assessment

Examples of Gears



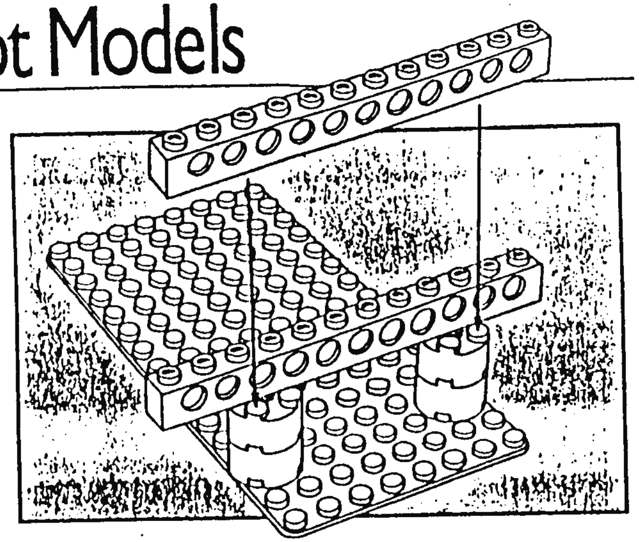
May be photocopied for student work. See pages 3–7.

Process and Inquiry Skills	Cause and effect Predicting Observing Hypothesizing	Measuring Problem Solving Inferring Communicating

Critical Thinking Skills

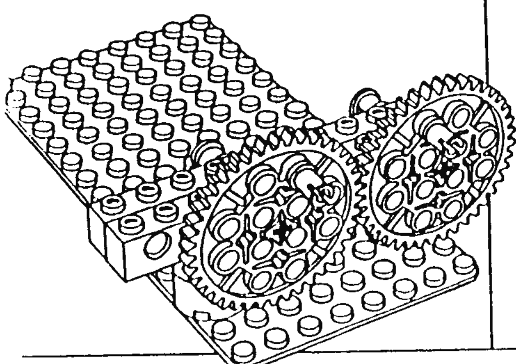
Gathering and recalling information
Understanding and interpreting data and information
Applying what is learned to solve problems in new situations
Analyzing a problem into its component parts
Synthesizing various aspects or components into a new whole to solve a problem
Evaluating one's own work

Gear Concept Models

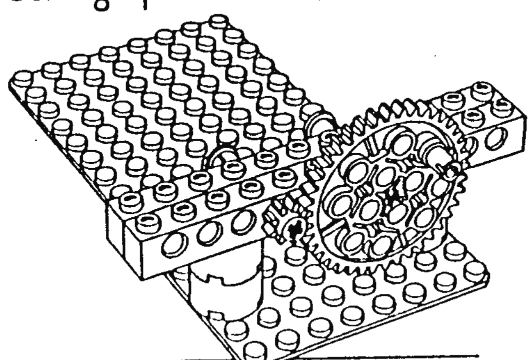


Build this base first

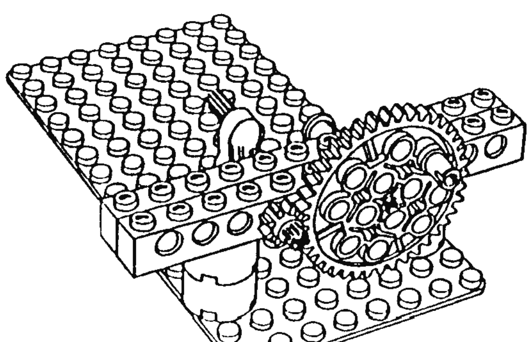
1 Direction of Rotation



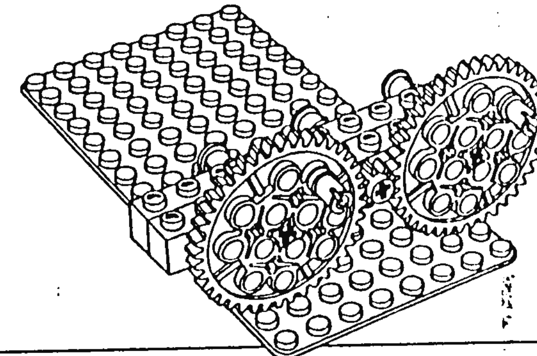
2 Gearing Up



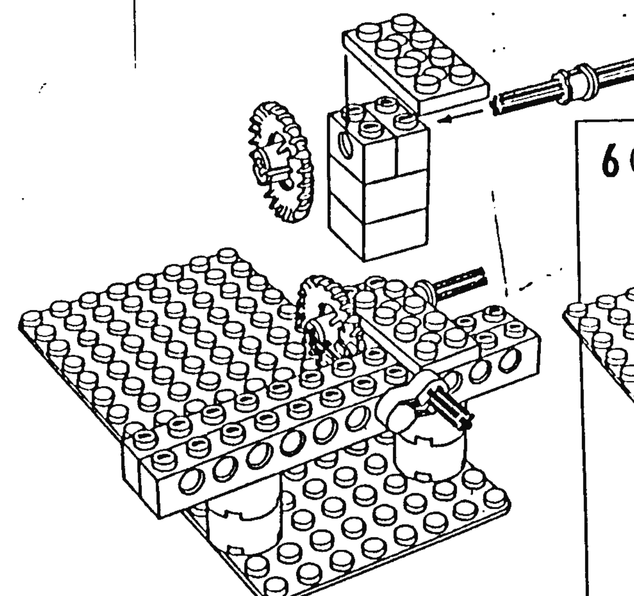
3 Gearing Down



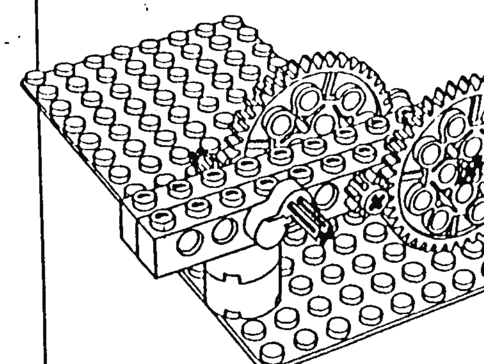
4 Idler Gearing



5 At An Angle



6 Compound Gearing



Project: Gears

Task:

Build the LEGO model shown on Page 1 of this handout.

Predict if the top will turn faster or slower than the handle.

What sort of gearing arrangement is this ?

Determine the gear ratio.

Modify the model to improve the situation.

What is the gear ratio now ?

Class Quiz - A NAME...

Answer the following questions:

1) The function of a gear is to
   - a) transfer force
   - b) change direction of rotary motion
   - c) maintain direction of rotary motion
   - d) both a) and b)
   - e) none of the above

2) The gear that you turn is called the
   - a) slave gear
   - b) follower gear
   - c) driver gear
   - d) primary gear
   - e) none of the above

3) Turning a 60 tooth gear meshed to a 12 tooth gear, the correct ratio is
   - a) gearing down 1/5
   - b) gearing up 1/5
   - c) gearing down 5/1
   - d) gearing up 5/1
   - e) none of the above

4) The function of an idler gear is to
   - a) look pretty
   - b) make gearing look impressive
   - c) make adjacent gears turn in the same direction
   - d) change motion through 90°
   - e) none of the above

5) A gear that can change rotary motion through 90° is called
   - a) an idler gear
   - b) a crown gear
   - c) a teara gear
   - d) a meshed 90° gear
   - e) an angular transfer gear

....

Tear off this section. Fill in the relevant details and hand to the project leader at the completion of the prac.

GROUP : ....

KIT NUMBER : ....

NAME : ....
....

FACULTY OF ENGINEERING
UNIVERSITY OF CAPE TOWN
ENGINEERING ONE
MEC104W & MAT104W
MECHANICAL ELECTRO-MECHANICAL MATERIALS

Levers Project

Day 1 - Classwork and Homework Task

Read the handout entitled **Levers in Action** and answer the following question to be handed in on **Day 2**. Be sure to include your name on the top right hand corner of the page that contains your answers.

Classify the type of lever shown below in the **front end loader**. Make sure that you understand how the various levers interact to control the bucket.

Day 2 - Classwork Task to be completed in groups of two

Build the LEGO model shown on page 1 of the handout **Levers in Action**.

How does this model use levers?

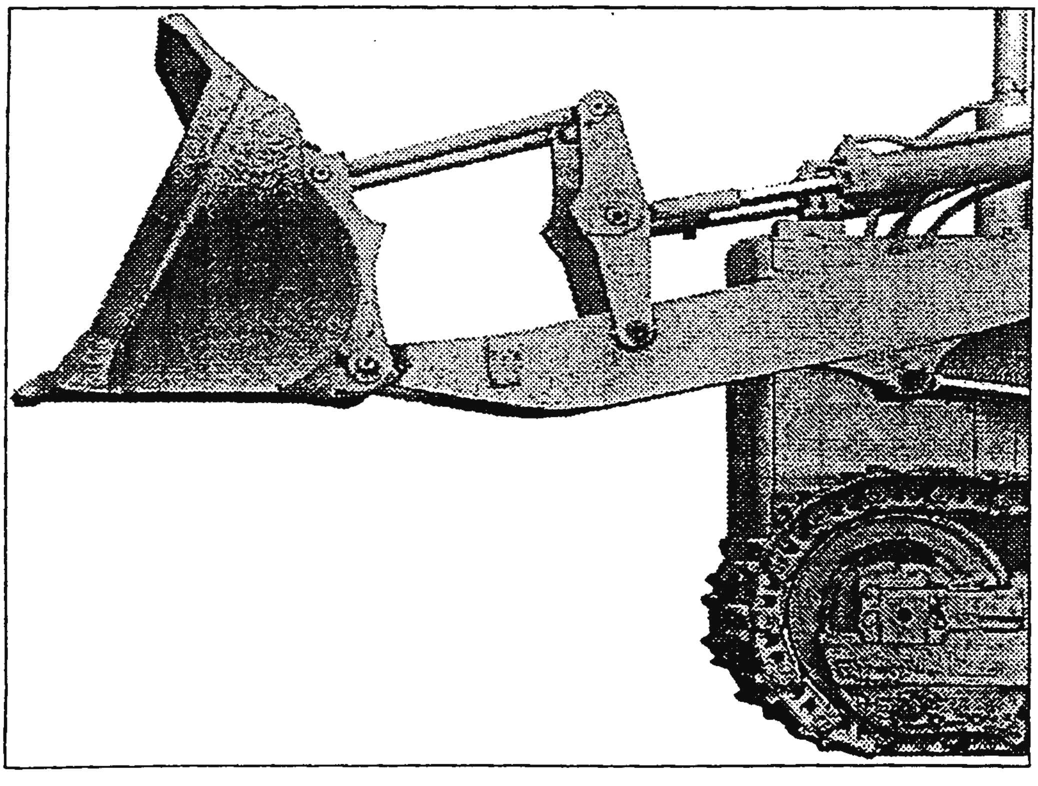
Predict what will happen if you place a small object on the tray. Test your prediction with several different objects..

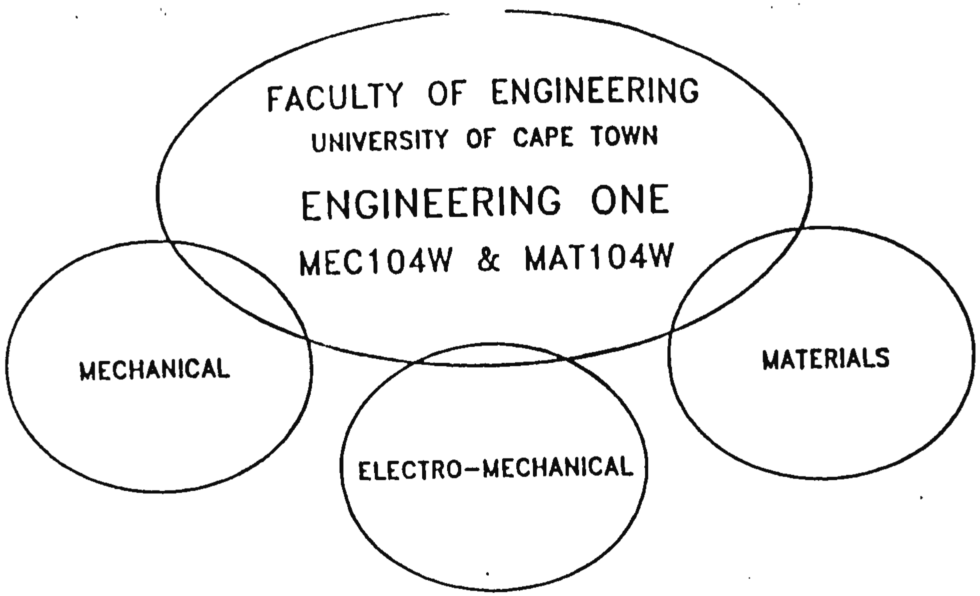
How can the principle used in this model be used in a practical and useful way?

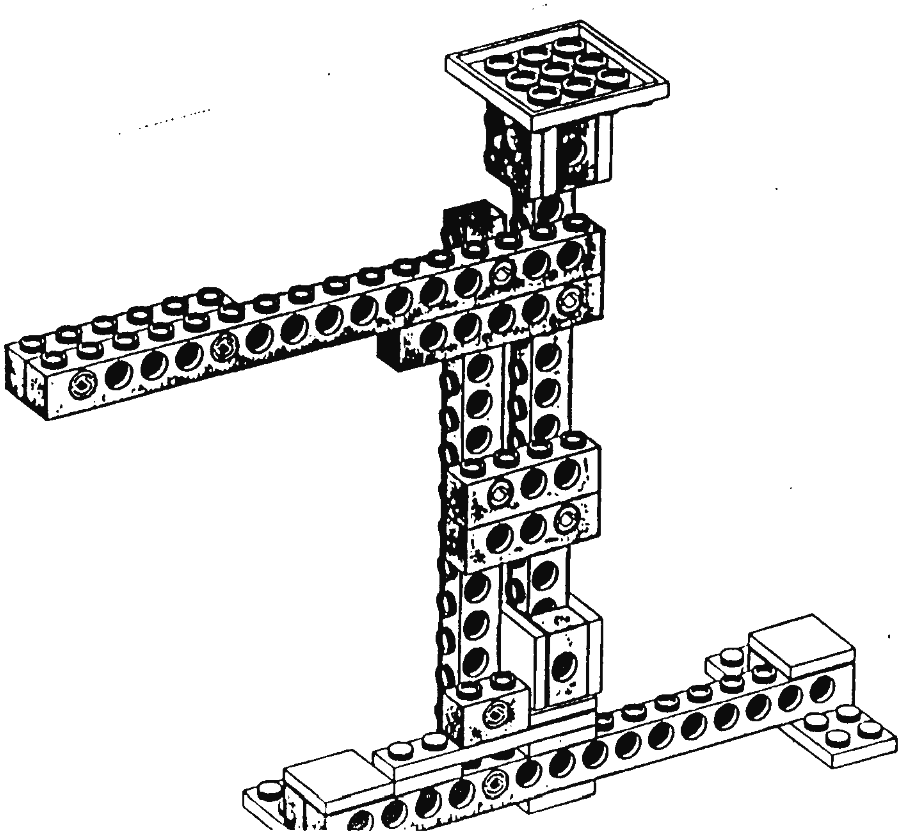
Modify the model to compare the weights of heavier objects.

Design cardboard scales to determine the weights of heavier objects.

Change the device into a press. How does the press use a lever?

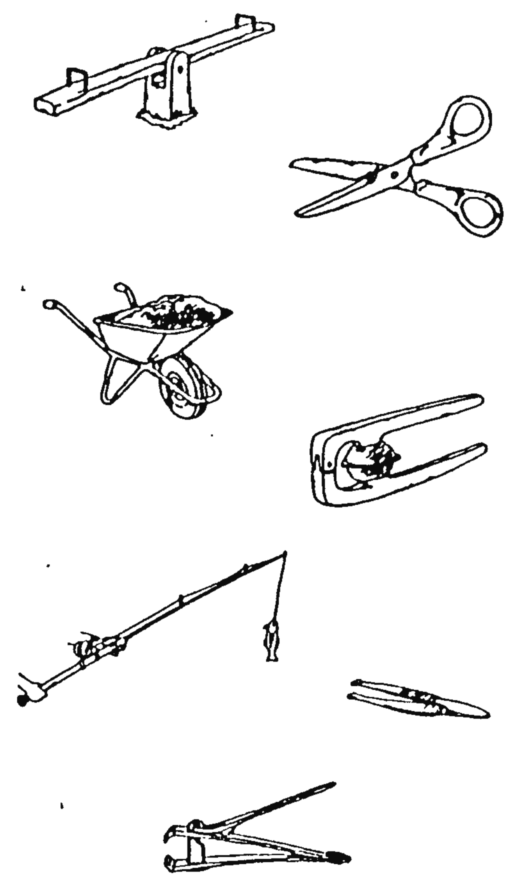






Levers in Action

A lever is a rod or arm that tilts around a pivot to produce useful motion. One of the simple machines, the lever can make work easier by amplifying motion or force, or by changing the direction of a force.



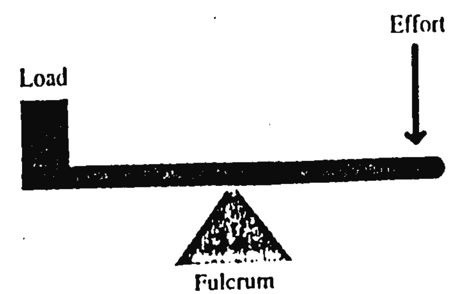
Here are some examples of levers or devices made up of levers.
The three classes of levers are discussed on pages 4–7.

- Seesaw (1st class)
- Scissors (connected 1st class)
- Wheelbarrow (2nd class)
- Nutcracker (connected 2nd class)
- Fishing rod (3rd class)
- Tweezers (connected 3rd class)
- Nail clippers (2nd class and connected 3rd class)

Photographs on the back of this booklet show other examples of how levers make work easier.

The diagram at the lower right shows the various parts of a lever in general.

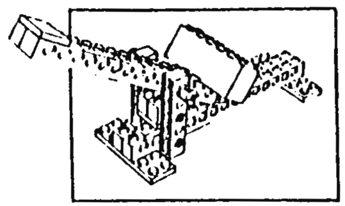
The load is moved by the **effort** (a push or pull) making the lever tilt about the **fulcrum** (pivot).



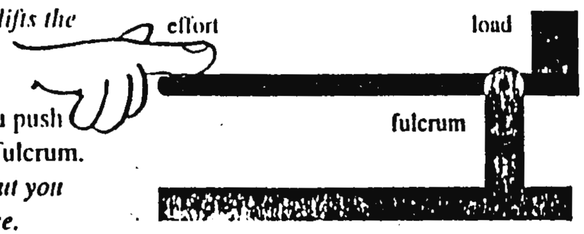
Familiarize yourself with levers by working through activities on pages 4–7. Photocopy the drawings on pages 16–17 for easy reference. You may wish to provide these activities to your students as well.

evers in Action

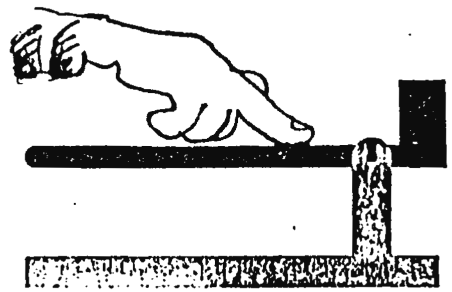
‑irst Class Lever



◆ Build the base and model 1 shown on ›age 16. Predict what happens when you ›ush down at the yellow tile on the handle.
As you push down (effort), the beam tilts about the axle (fulcrum) and lifts the weighted brick (load).

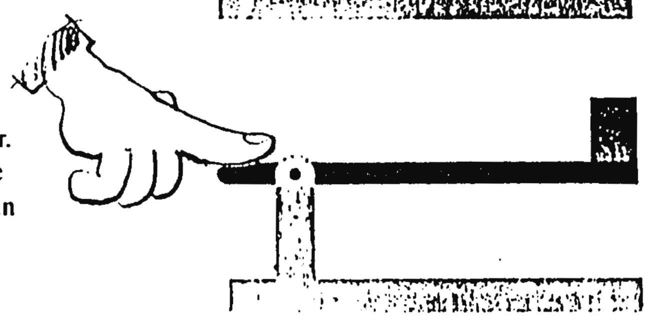


■ Predict what happens when you push down on the handle closer to the fulcrum.
You also lift the weighted brick, but you have to push down with more force.



Main Ideas:

A first class lever has the fulcrum between the effort and the load. It is easier to lift a load with a first class lever if you push down at a point far from the fulcrum.

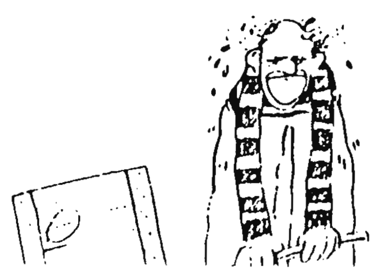


Additional Information:

First class levers are often used to change the direction of forces. A seesaw is an example of a first class lever. First class levers can also be connected to each other, as in a pair of scissors.

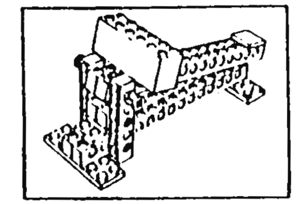
Extension:

Switch the places of the yellow tile and the weighted brick. Repeat the above investigation. Then change the location of the brick along the beam.
(It takes a great effort to raise the brick, because you are pushing down so close to the fulcrum. However, as the load is also moved closer to the fulcrum, the effort needed becomes less.)

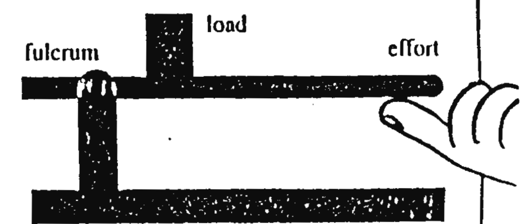


Levers in Action

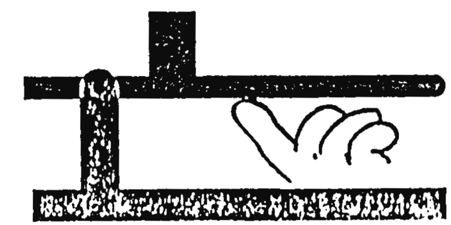
Second Class Lever



◆ **Build model 2 shown on page 17. Predict what happens when you push up at the yellow tile on the handle.**
As you push up (effort), the beam tilts about the axle (fulcrum) and lifts the weighted brick (load).

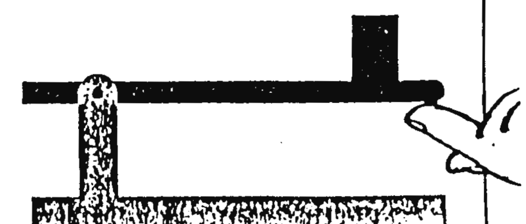


■ Predict what happens when you push up on the handle closer to the fulcrum.
You also lift the weighted brick, but you have to push up with more force.



Main Ideas:

A second class lever has the load between the fulcrum and the effort. The effort will always be less than the load. The farther from the fulcrum you apply the effort, the easier it is to lift the load.

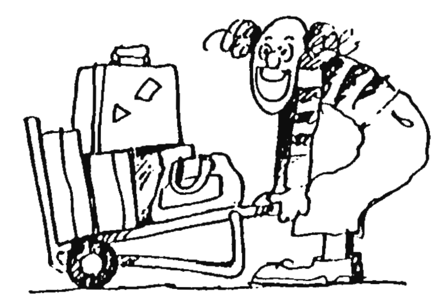


Additional Information:

Second class levers are often used to lift heavy loads or to apply large forces.
A wheelbarrow is a typical example of a second class lever. Second class levers can also be connected to each other, as in a nutcracker.

Extension:

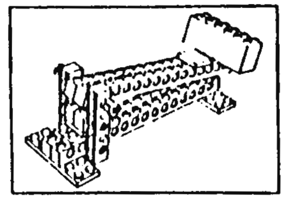
Move the weighted brick closer to the yellow tile and repeat the above investigation.
(It takes more effort to raise the load as it is moved further from the fulcrum. When the load is closest to the fulcrum, it can be lifted with the least effort.)

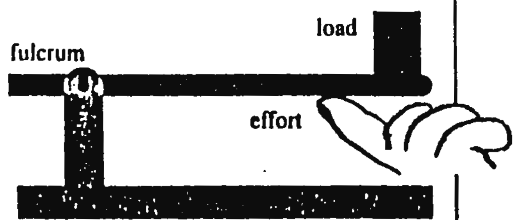


.evers in Action

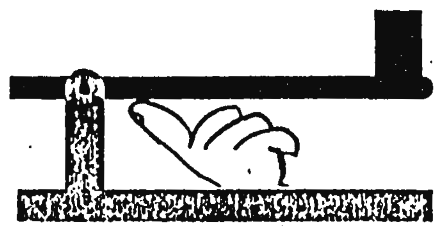
Third Class Lever

◆ Build model 3 shown on page 17.
Predict what happens when you push up on the handle at the yellow tile.
As you push up (effort), the beam tilts about the axle (fulcrum) and lifts the weighted brick (load).





■ Predict what happens when you push up on the handle closer to the fulcrum.
You also lift the weighted brick, but you have to push up with more force.



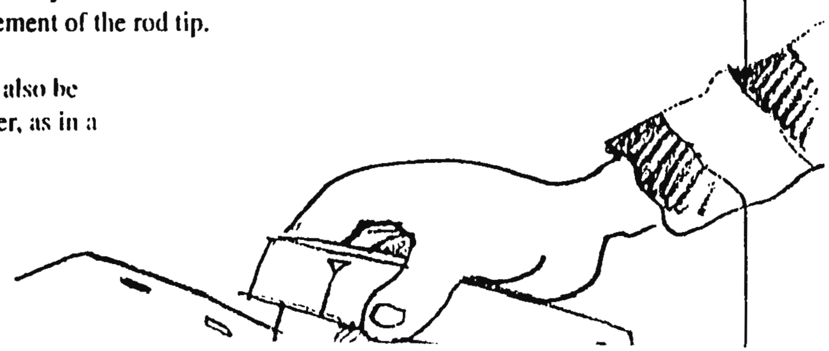
Main Ideas:

A third class lever has the effort between the fulcrum and the load. The farther from the fulcrum the load is located, the more effort you will have to apply to lift it.

Additional Information:

Although third class levers can be used to lift loads or apply specific forces, as in a stapler, they are also used to amplify motion. A fishing rod is a typical example. The fulcrum is the hand at the end of the rod handle. A small motion of your other hand produces a large movement of the rod tip.

Third class levers can also be connected to each other, as in a pair of tweezers.



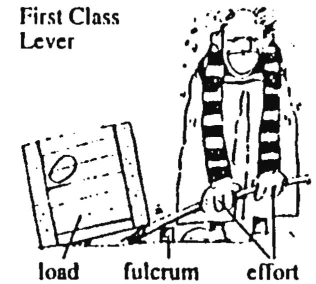
Levers in Action

Lever Summary

First Class Levers

The fulcrum is between the effort and the load. A seesaw is an example of a simple first class lever. A pair of scissors is an example of two connected first class levers.

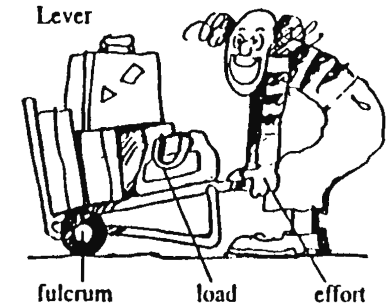
First Class Lever



Second Class Levers

The load is between the fulcrum and the effort. A wheelbarrow is an example of a simple second class lever. A nutcracker is an example of two connected second class levers.

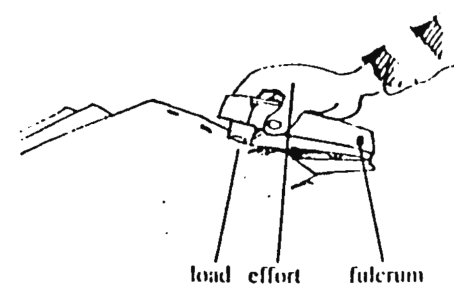
Second Class Lever



Third Class Levers

The effort is between the fulcrum and the load. A stapler or a fishing rod is an example of a simple third class lever. A pair of tweezers is an example of two connected third class levers.

Third Class Lever



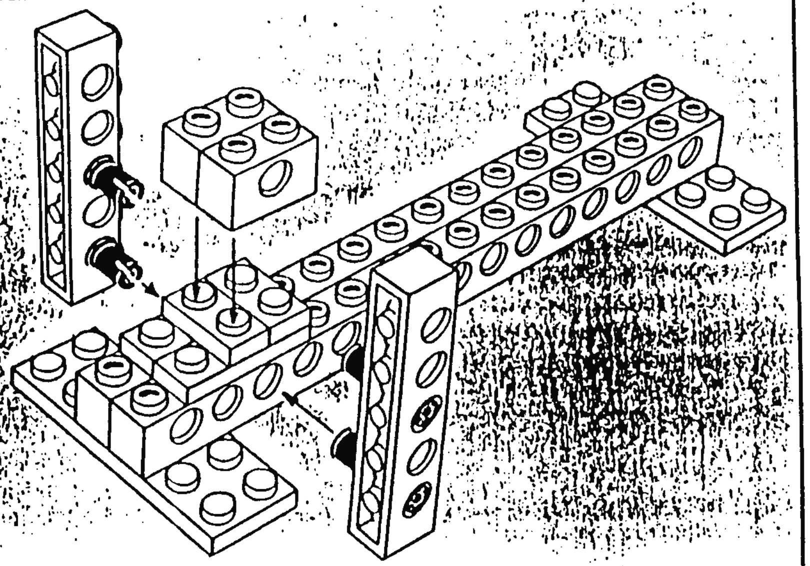
Force and Effort

To lift a load with the least effort:

■ place the load as close to the fulcrum as possible.

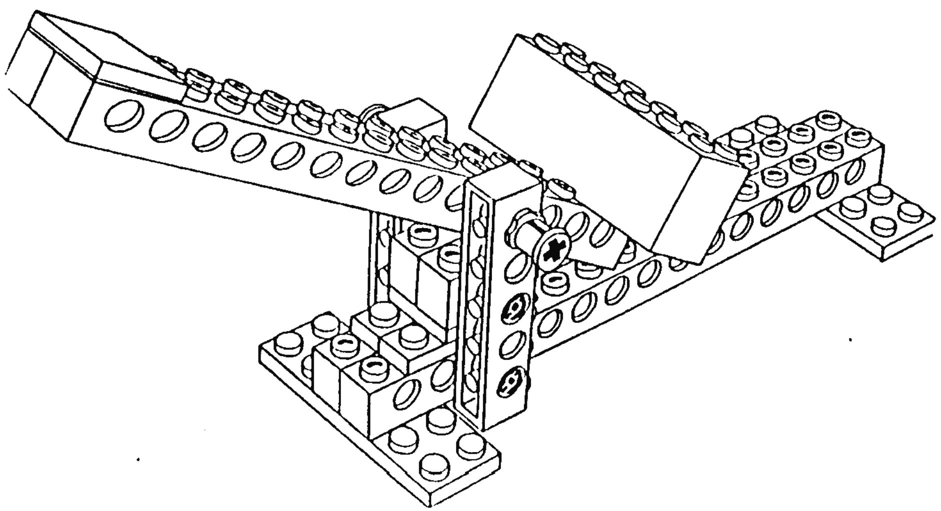
■ apply the effort as far from the fulcrum as possible.

ɜver Concept Models

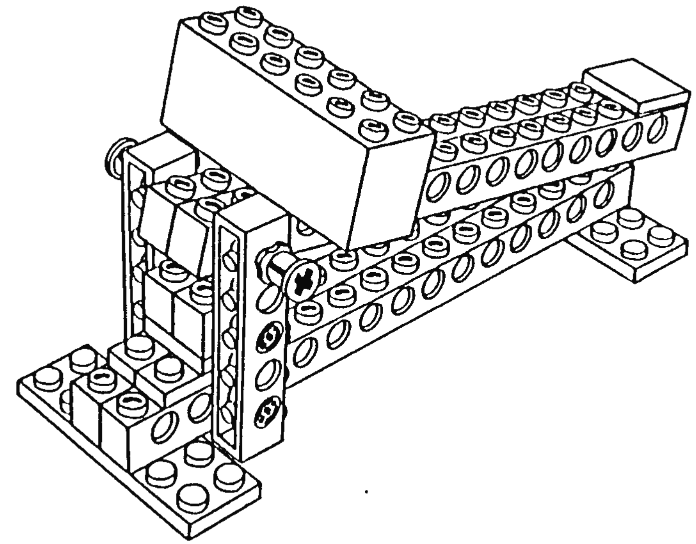


ɪuild this
ılse first
ɪote the use
ɪf black
dırector
ɪegs:

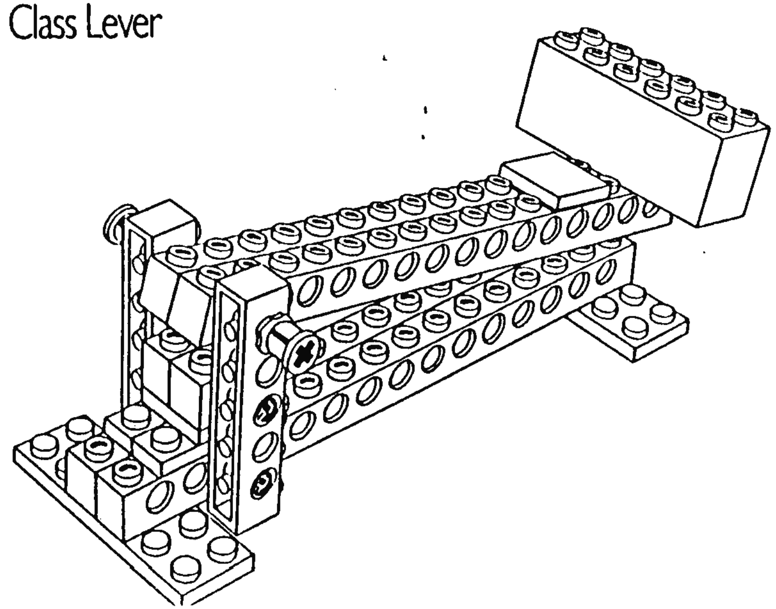
ɪrst Class Lever



2 Second Class Lever



3 Third Class Lever



Project: Levers

Task:

Build the LEGO model shown on Page 1 of this handout.

How does this model use levers ?

Predict what will happen if you place a small object on the tray. Test your prediction with several different objects.

How can the principle used in this model be used in a practical and useful way ?

Modify the model to compare the weights of héavier objects.

Design cardboard scales to determine the weights of heavier objects.

Change the device into a press.

How does the press use a lever ?

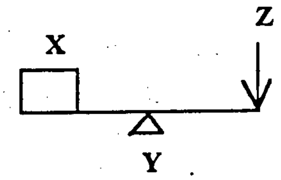
Class Quiz - A

NAME...

1) A lever can

   a) amplify motion
   b) amplify a force
   c) change the direction of an applied force
   d) a) and c)
   e) all of the above

2) Which letter descrides the situation in the sketch ?



	X	Y	Z
a)	effort	load	fulcrum
b)	fulcrum	load	effort
c)	load	fulcrum	effort
d)	load	effort	fulcrum
e)	effort	fulcrum	load

3) Which letter in Question 2) above represents a **FIRST CLASS** lever with respect to the sketch below ?

X ______________________ Z
Y

4) Which letter in Question 2) above represents a **SECOND CLASS** lever with respect to the sketch below ?

X ______________________ Z

5) Which letter in Question 2) above represents a **THIRD CLASS** lever with respect to the sketch below ?

X ______________________ Z

....

Tear off this section. Fill in the relevant details and hand to the project leader at the completion of the prac.

GROUP : ....

KIT NUMBER : ....

NAME : ....
....

"Introduction to Mechanics" Workshop Problem Sheet

(Total of 1 page)

LIST OF PROJECTS

First Semester
First cycle of projects

1. Wheel Balancing
2. Pulley experiments
3. construct a model helicopter (electric)*
4. construct a model racing car (electric)*
5. construct a mechanical hoist (pneumatic)*
6. construct a mechanical claw (pneumatic)*

* Assembly manuals provided

Second semester
Second cycle of projects.

1. mechanical crane (electric)
2. Industrial mixer (pneumatic)

Third cycle of projects.
Preliminary rounds in the competition.

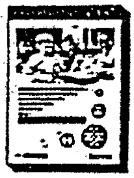
1. The fastest car
2. A vertical winch to lift the largest mass
3. A car to push the largest mass
4. A sideways winch to pull the largest mass

The winning team from the preliminary rounds competed in the finals during which they had to construct a mechanism to lift the largest mass as high as possible from the table.

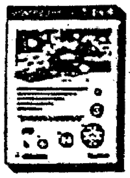
Appendix 4 Lego sensor and electrical component specifications

(Total of 3 pages)

1031
E . ding.
c. :ds from
A. . no.
1030

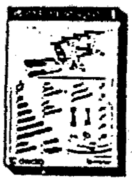


1033
Building
cards from
Art no.
1032

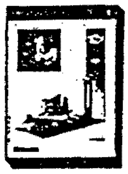


9873
Building
cards from
Art no.
9700

9877
E ''ding
c ;ds from
A. 1o.
9604



9879
Building
cards from
Art no.
9702

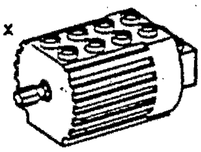


Γ'ectrical components

4.5

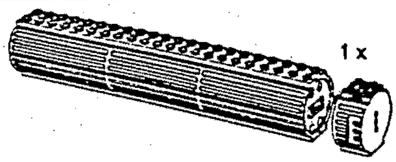
9

9859
4 volt motor



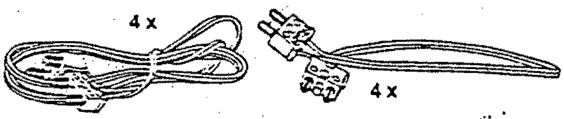
1 x

9860
4.5 volt battery box
Two-way switch



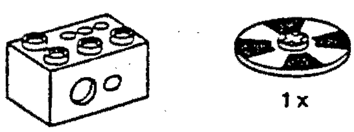
1 x

9861
4. volt
co..necting
leacs (25 and
75 cm long)



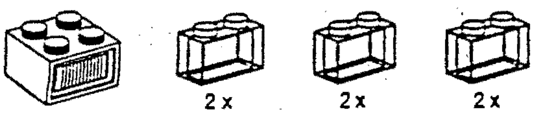
4 x
4 x

9865
4. olt
Op. sensor
Counting disc



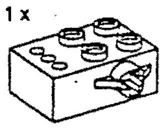
1 x
1 x

9866
4.5 volt
lig bricks
tr. sparent
coloured
bricks



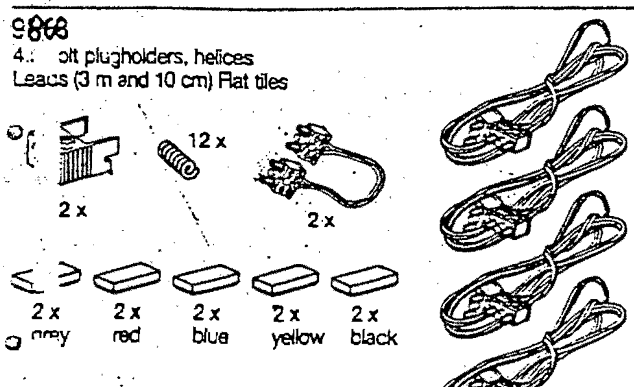
6 x
2 x red
2 x yellow
2 x green

9867
4.5 volt
touch sensor



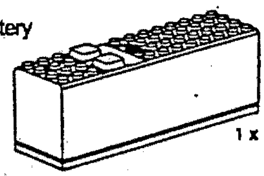
1 x

9868
4.: olt plugholders, helices
Leacs (3 m and 10 cm) Flat tiles



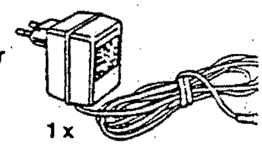
2 x
12 x
2 x
2 x grey
2 x red
2 x blue
2 x yellow
2 x black
1 x grey/black
1 x grey/red
1 x grey/blue
1 x grey/yellow

9831
9 volt battery box, including two-way switch



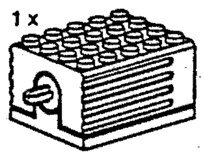
1 x

9833
9 volt trans-former



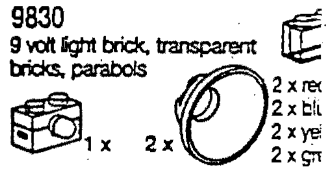
1 x

9883
9 volt motor



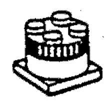
1 x

9830
9 volt light brick, transparent bricks, parabols



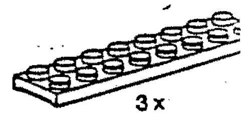
1 x
2 x
2 x red
2 x blu
2 x yel
2 x gr

9885
9 volt sound element



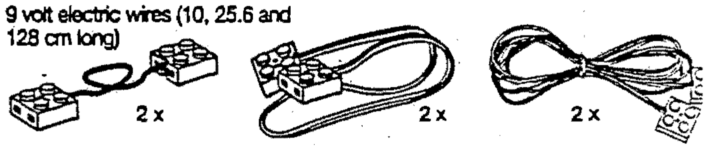
1 x

9886
9 volt electric plates



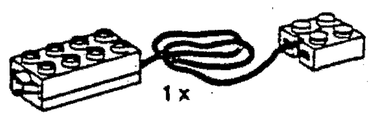
3 x

9887
9 volt electric wires (10, 25.6 and 128 cm long)



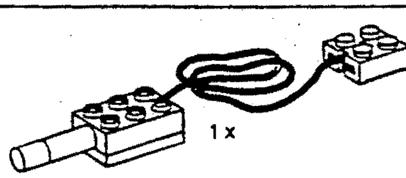
2 x
2 x
2 x

9888
9 volt touch sensor



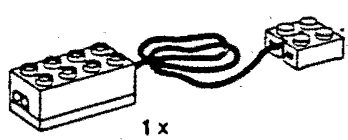
1 x

9889
9 volt temperature sensor



1 x

9890
9 volt light sensor



1 x

9891
9 volt angle sensor

Yellow Sensors

The yellow sensors do not require power to work and should only be connected to the yellow input ports labelled one to four on the LEGO DACTA Interface B.
There are two yellow sensors, the temperature sensor and the touch sensor.

Temperature Sensor (6)

The temperature sensor is an analogue sensor. It has a range of -20 to +50° C (6a).
The LEGO DACTA software allows the temperature sensor to be calibrated to allow measurement in both Celsius and Fahrenheit.

Touch Sensor (7)

The touch sensor is a digital switch. The LEGO DACTA software can detect when the button is pressed. It can count multiple (7a) or single presses (7b).

Safety Notes

The set is designed for safe use in the classroom environment.
The output devices should only be used with a 9 volt electrical supply.
The 9V connector should not be placed in water or any other liquids.

Technical References

Output devices

	Voltage	Power Consumption	Revolutions
motor without load	9V	app. 40 mA	appr. 4200 RPM
motor with normal load	9V	app. 160 mA	appr. 1600 RPM
lamp	9V	app. 30 mA	
sound element	9V	app. 0,2 mA	

Input devices

	Type	Sensor Range	Measuring	Accuracy	Range Limits
touch sensor	non powered	digital			
temperature sensor	non powered	analogue	-20° to +50°C	± 1°C at 25°C	appr. ±1°C
angle sensor	powered	digital	16 positions per revolution		max. 500 RPM w/function
[illegible]	[illegible]	[illegible]	[illegible]		

9701

Directions for use

LEGO dacta®

Introduction

Congratulations on purchasing the new LEGO DACTA® Technology Models for Computer Control.

The set consists of:

- Building elements - including sensors, motors, lamps and sound element.
- Building instructions for seven models.
- This instruction manual.

The set is designed to support the LEGO DACTA Interface B (item 9751). A curriculum support pack based on these models is also available from your normal LEGO DACTA educational supplier. The pack includes LEGO DACTA software, pupil work cards and teachers guides.

Models

The set includes step by step building instructions for the following models:

- Greenhouse
- Vending machine
- Car testing station
- A joystick controlled wheel chair
- Scanner
- Conveyor belt
- Robot arm

Seven further models are also illustrated. These include a fan, traffic lights, bar code reader, programmable disc, robot arm, programmable robot arm, two motor buggy.

Output Devices

The 9701 set includes the following 9 volt output devices:

- Motors ①
- Lamps ②
- Sound element ③

These output devices should be connected to output ports labelled A-H on the LEGO DACTA Interface B. They can also be connected to the test port. In both cases they should be connected using the 9V connectors provided with the 9701 kit.

The LEGO DACTA software provides the following control functions;

Motors

- Turn on anti-clockwise/off (1a)
- Turn on clockwise/off (1b)
- Change direction (1c)
- Control the power level (1d)

Lamp

- Turn on/off (2a)
- Flashing interval (2b)
- Intensity (2c)

Sound element

- Turn on/off (3a)
- Make two types of sound (3b)

Input Devices

The 9701 construction kit includes the following input devices:

- A light sensor ④
- An angle sensor ⑤
- A temperature sensor ⑥
- A touch sensor ⑦

There are two types of sensors, blue and yellow.

Blue Sensors

There are two kinds of blue sensors, a light sensor and an angle sensor. These sensors require power to operate correctly and should only be connected to the blue input ports. These are numbered 5-8 on the LEGO DACTA Interface B

Light Sensor ④

This is a very versatile sensor which,when used in conjunction with the LEGO DACTA Interface B can;

- Count, by measuring variations in light levels (4a)
- Work as a light barrier (4b)
- Read a LEGO "barcode", grey scale sorting (4c)

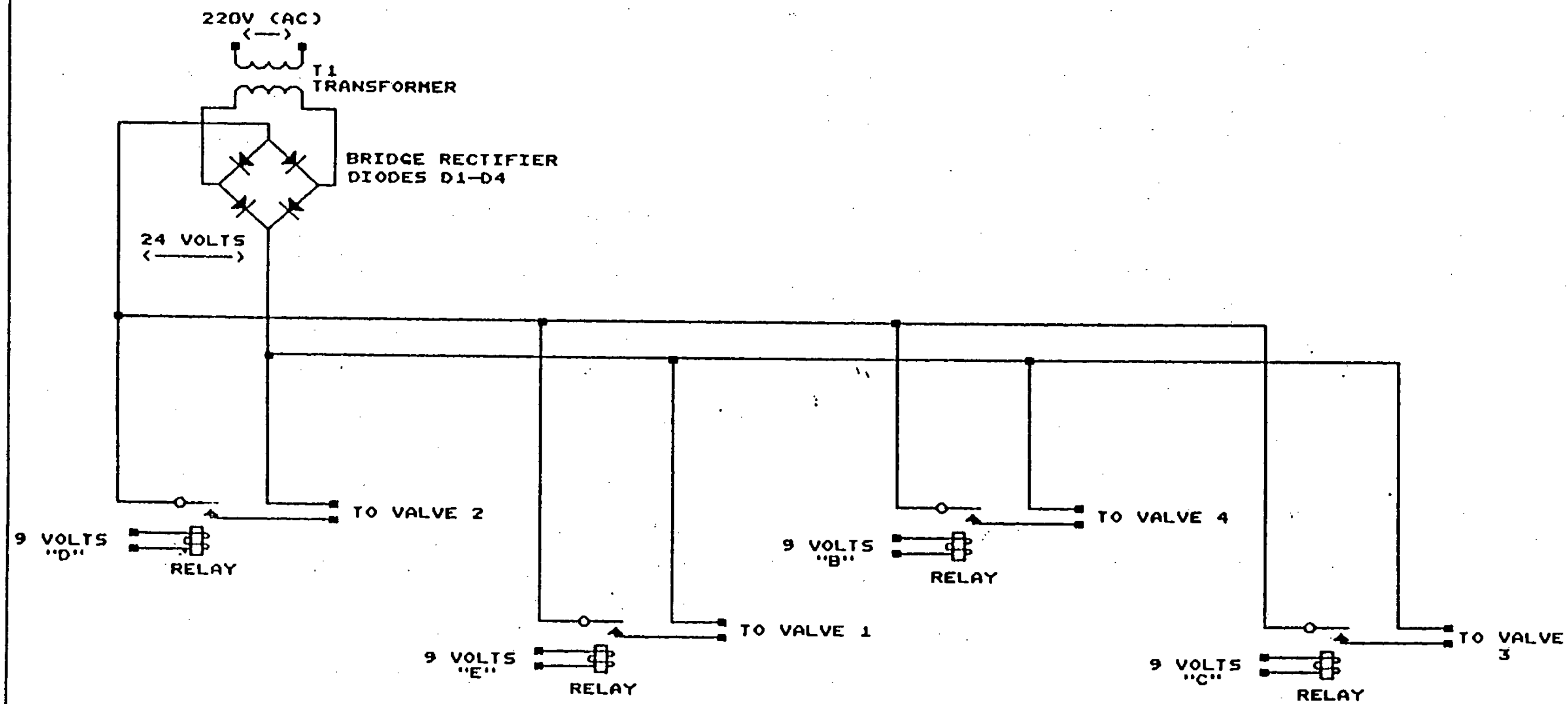
Angle Sensor ⑤

The angle sensor which,when used in conjunction with the LEGO DACTA interface B can;

- Identify direction (5a)
- Count the number of rotations (5b)
- Record and measure angles (5c)

Appendix 5 Conveyor Model: Electronic Diagram

(Total of 1 page)

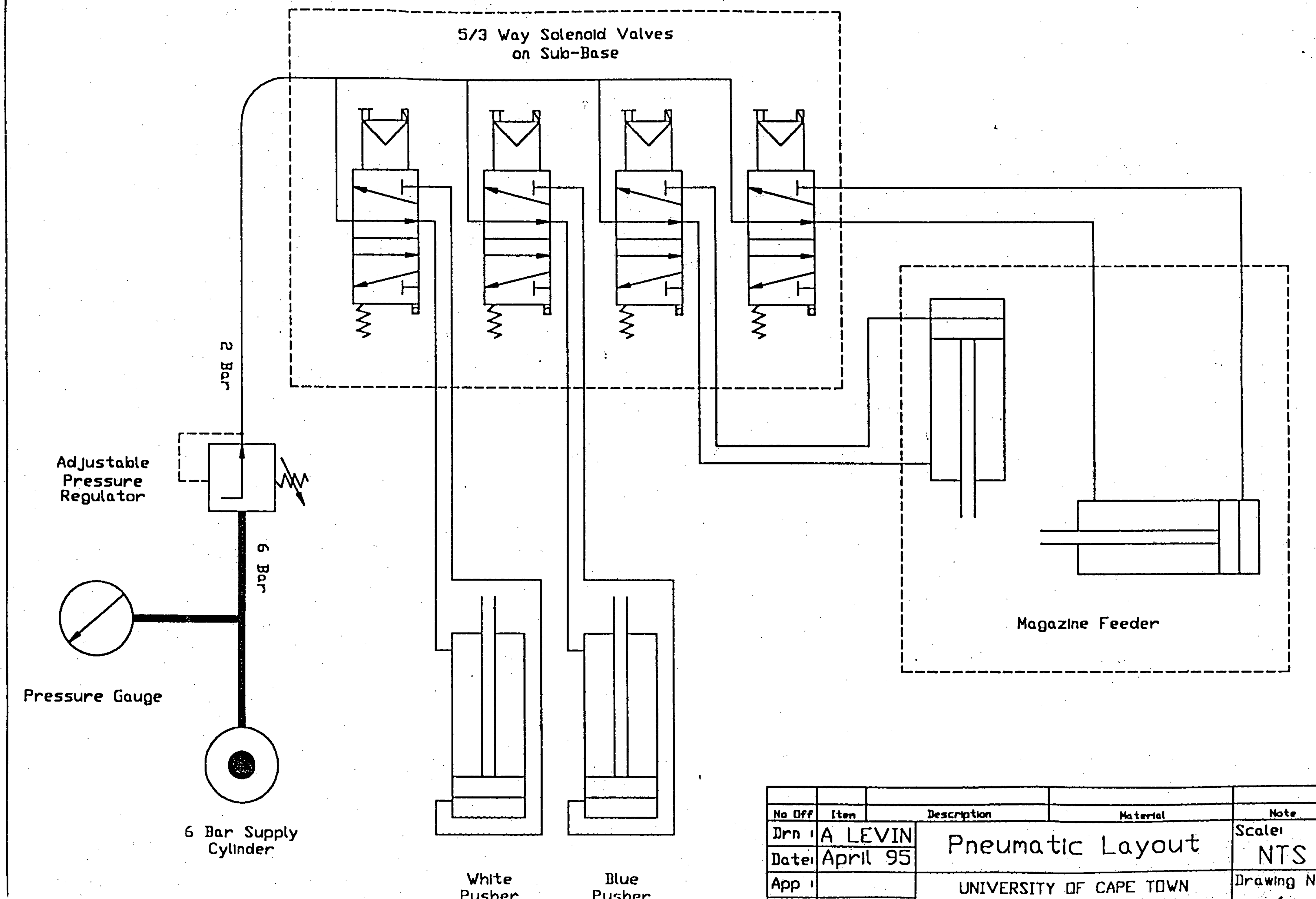


NOTE: 9 VOLTS D.C. FROM LEGO INTERFACE "B"

MECHANICAL ENGINEERING DEPARTMENT		
Title UNIVERSITY OF CAPE TOWN		
Size A	Document Number 220 V-AC TO 24 V-DC CONVERTOR	REV 1
Date: September 3, 1995	Sheet 1 of 1	

Appendix 6 Conveyor Model: Pneumatic Diagram

(Total of 3 pages)



No Off	Item	Description	Material	Note
Drn :	A LEVIN	Pneumatic Layout		Scale: NTS
Date:	April 95			
App :		UNIVERSITY OF CAPE TOWN		Drawing No: 1

Pneumatics and Logic Symbols

to ISO 1219

Energy Transmission

- Working line
- Flexible line
- Pressure source
- Electric line
- Control line
- Exhaust line
- Line connection (rigid)
- Line cross-over
- Exhaust point
- Silencer
- Drain without pipe connection
- Drain with pipe connection
- Air connection point, closed
- Air connection point with connected line
- Quick release coupling, coupled
- Quick release coupling with mechanically opened nonreturn valves, coupled
- Quick release coupling, uncoupled, line open
- Quick release coupling, uncoupled, line closed
- Rotary connection with one path
- Rotary connection, e.g. with three paths
- Pneumatic capacitor
- Filter
- Water trap, manual
- Water trap with automatic drain
- Filter with water trap
- Filter with water trap, automatic
- Cooler
- Cooler
- Dryer
- Lubricator
- Service unit (simplified graph)

Control Methods

Mechanical Components

- Detent
- Over-centre device
- Shaft rotational movement in one direction
- Shaft rotational movement in two directions
- Locking device (*control method for releasing the locking device)
- Hinge joint with extended lever
- Hinge joint, simple
- Hinge with fixed pivot

Pressure Controls

- Direct by application of pressure
- Pressure centred
- Differential pressure actuation
- Spring centred
- Indirect by application of pressure

Combined Controls

- By solenoid and pilot valve
- By solenoid or pilot valve
- By solenoid or manual operation with return spring
- General, *explanatory symbol

Manual Controls

- General
- Button
- Lever
- Pedal

Mechanical Controls

- Plunger
- Spring
- Roller
- Roller with idle returr

Electrical Controls

- Solenoid with one effective winding
- Solenoid, e.g. with two windings acting in opposition
- Electric motor with continu-ous rotary motion
- Electric stepping mctc

Special Controls

- By application of pressure through pressurized amplifier*
- By application pressure, typ: control produc alternating behaviour*

* = not standardized, proposal

- Mechanical return
- Actuated in starting positi

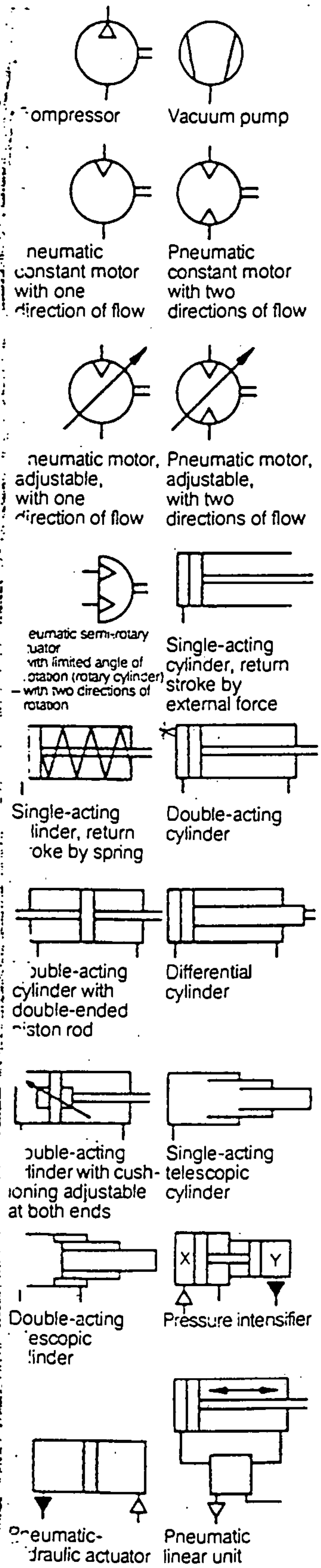
050170 GB

FESTO

Pneumatics and Logic Symbols

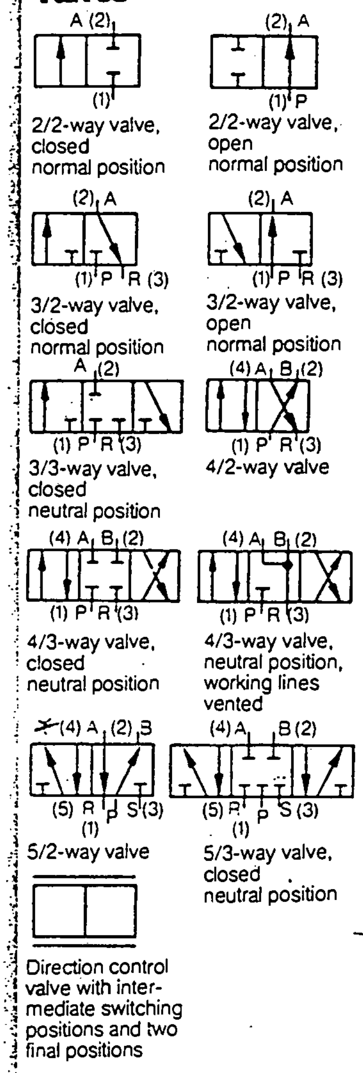
to ISO 1219

Energy Conversion

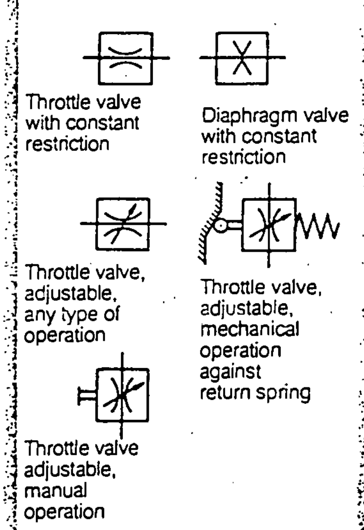


Valves

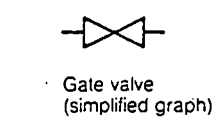
Directional Control Valves



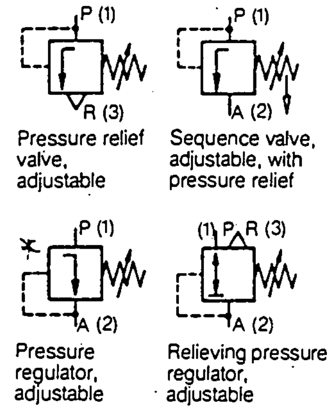
Flow Control Valves



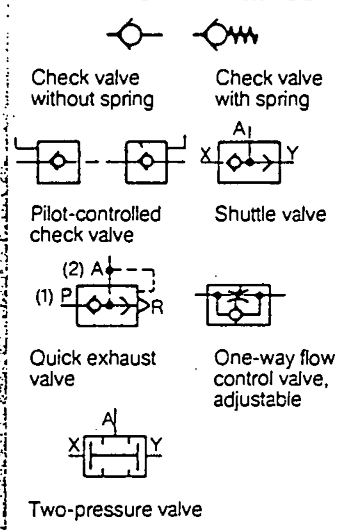
Gate Valve



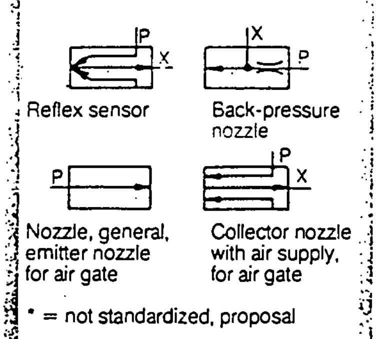
Pressure Control Valves



Non-Return Valves



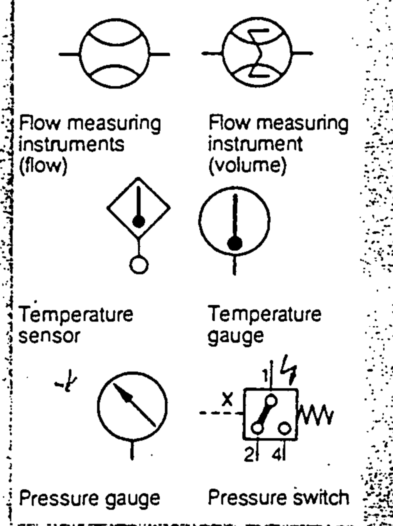
Special Symbols*



* = not standardized, proposal

Other Symbols

Other Devices

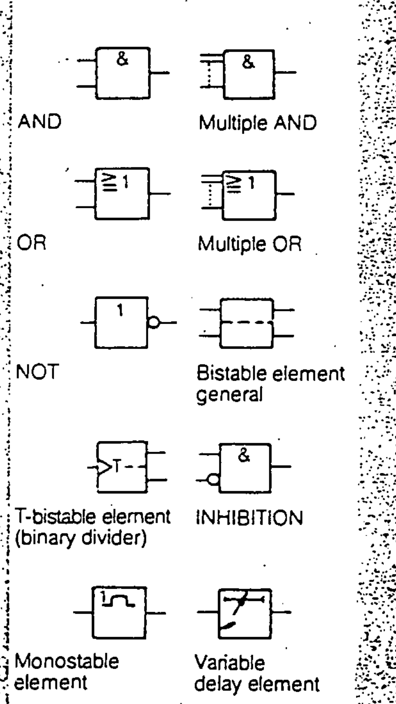


ISO STANDARD 5599/II

Designation of Connections

A, B, C (2, 4, 6)	working lines
P (1)	compressed air connection
R, S, T (3, 5, 7)	drain, exhaust points
L	leakage fluid
Z, Y, X (12, 14, 16)	control lines

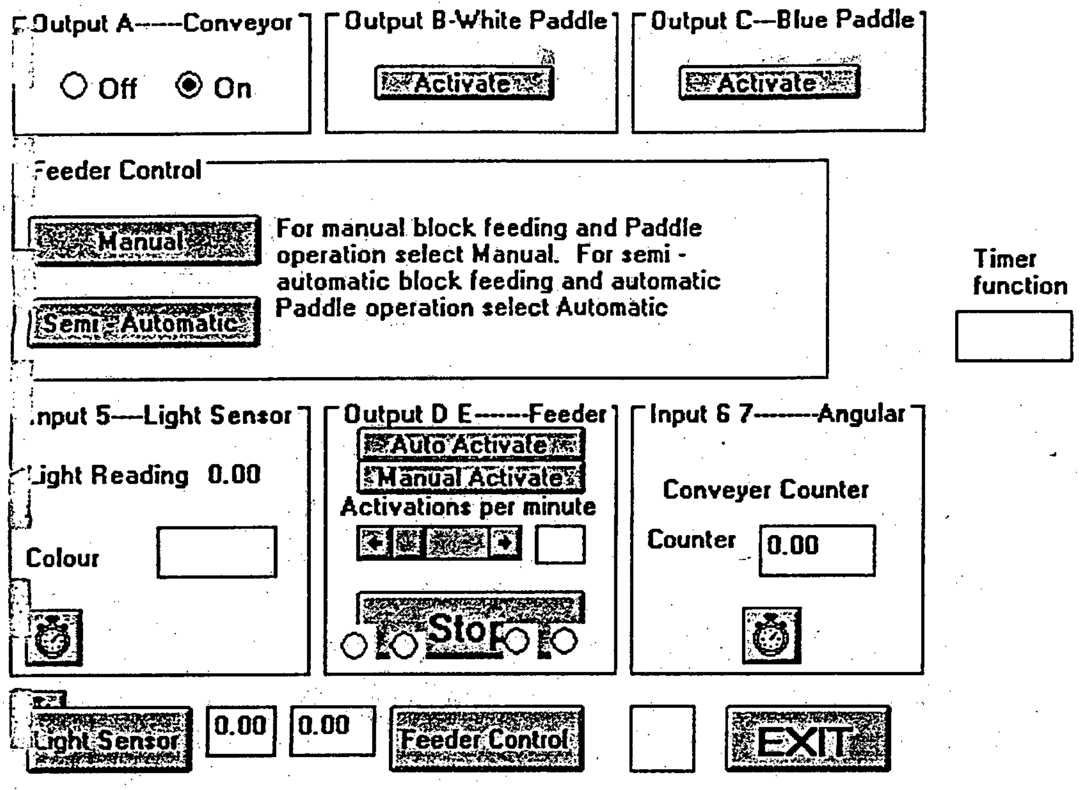
Logic Symbols



FESTO

Appendix 7 CONVEYOR.EXE Programme Code

(Total of 22 pages)


Output A-----Conveyor
Off
On
Output B-White Paddle
Activate
Output C--Blue Paddle
Activate
Feeder Control
Manual
Semi - Automatic
For manual block feeding and Paddle operation select Manual. For semi - automatic block feeding and automatic Paddle operation select Automatic
Timer function
Input 5---Light Sensor
Light Reading 0.00
Colour
Output D E------Feeder
Auto Activate
Manual Activate
Activations per minute
Stop
Input 6 7--------Angular
Conveyer Counter
Counter
0.00
Light Sensor
0.00
0.00
Feeder Control
EXIT

```
VERSION 2.00
Begin Form Form1
   BackColor       =   &H00FFFFFF&
   Caption         =   "Conveyor Manual Opperation"
   ClientHeight    =   6405
   ClientLeft      =   210
   ClientTop       =   1710
   ClientWidth     =   8070
   FontBold        =   -1  'True
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   9.75
   FontStrikethru  =   0   'False
   FontTransparent =   0   'False
   FontUnderline   =   0   'False
   Height          =   7095
   Icon            =   (Icon)
   Left            =   150
   LinkTopic       =   "Form1"
   ScaleHeight     =   6405
   ScaleWidth      =   8070
   Top             =   1080
   Width           =   8190
   Begin TextBox Text18
      Height          =   375
      Left            =   7080
      TabIndex        =   15
      Top             =   2400
      Visible         =   0   'False
      Width           =   855
   End
   Begin TextBox Text10
      Height          =   495
      Left            =   4680
      TabIndex        =   31
      Top             =   5280
      Visible         =   0   'False
      Width           =   495
   End
   Begin Frame Frame6
      Caption         =   "Feeder Control"
      Height          =   1695
      Left            =   120
      TabIndex        =   27
      Top             =   1200
      Visible         =   0   'False
      Width           =   6015
      Begin TextBox Text9
         BorderStyle     =   0   'None
         ForeColor       =   &H00FF0000&
         Height          =   975
         Left            =   1920
         MultiLine       =   -1  'True
         TabIndex        =   30
         Text            =   "For manual block feeding and Paddle operation s
         Top             =   480
         Width           =   3495
      End
      Begin CommandButton Command9
         Caption         =   "Semi - Automatic"
         Height          =   375
         Left            =   120
         TabIndex        =   29
         Top             =   1080
         Width           =   1695
```

```
      End
      Begin CommandButton Command8
         Caption         =   "Manual"
         Height          =   375
         Left            =   120
         TabIndex        =   28
         Top             =   480
         Width           =   1695
      End
   End
   Begin CommandButton Command7
      Caption         =   "Feeder Control"
      Height          =   495
      Left            =   2880
      TabIndex        =   21
      Top             =   5280
      Visible         =   0   'False
      Width           =   1455
   End
   Begin TextBox Text8
      Height          =   375
      Left            =   2160
      TabIndex        =   19
      Text            =   "0.00"
      Top             =   5280
      Visible         =   0   'False
      Width           =   615
   End
   Begin TextBox Text7
      Enabled         =   0   'False
      Height          =   375
      Left            =   1560
      TabIndex        =   20
      Text            =   "0.00"
      Top             =   5280
      Visible         =   0   'False
      Width           =   495
   End
   Begin Timer Timer3
      Enabled         =   0   'False
      Interval        =   1000
      Left            =   120
      Top             =   5160
   End
   Begin CommandButton Command3
      Caption         =   "Light Sensor"
      Height          =   495
      Left            =   240
      TabIndex        =   22
      Top             =   5280
      Width           =   1215
   End
   Begin Frame Frame5
      Caption         =   "Input 6 7--------Angular"
      Height          =   2055
      Left            =   4680
      TabIndex        =   14
      Top             =   3000
      Width           =   2175
      Begin Timer Timer1
         Interval        =   250
         Left            =   840
         Top             =   1560
      End
      Begin TextBox Text6
```

```
         Height          =   375
         Left            =   960
         TabIndex        =   18
         Text            =   "0.00"
         Top             =   960
         Width           =   855
      End
      Begin Label Label8
         Caption         =   "Counter"
         Height          =   255
         Left            =   120
         TabIndex        =   17
         Top             =   960
         Width           =   735
      End
      Begin Label Label7
         Caption         =   "Conveyer Counter"
         Height          =   255
         Left            =   240
         TabIndex        =   16
         Top             =   600
         Width           =   1695
      End
   End
   Begin Frame Frame4
      Caption         =   "Output D E-------Feeder"
      Height          =   2055
      Left            =   2400
      TabIndex        =   13
      Top             =   3000
      Visible         =   0   'False
      Width           =   2175
      Begin CommandButton Command10
         BackColor       =   &H00FFFFFF&
         Caption         =   "Stop"
         FontBold        =   -1  'True
         FontItalic      =   0   'False
         FontName        =   "MS Sans Serif"
         FontSize        =   13.5
         FontStrikethru  =   0   'False
         FontUnderline   =   0   'False
         Height          =   495
         Left            =   240
         TabIndex        =   3
         Top             =   1440
         Visible         =   0   'False
         Width           =   1695
      End
      Begin OptionButton Option7
         Caption         =   "Option7"
         Height          =   255
         Left            =   1680
         TabIndex        =   38
         Top             =   1680
         Visible         =   0   'False
         Width           =   255
      End
      Begin OptionButton Option6
         Caption         =   "Option6"
         Height          =   255
         Left            =   1320
         TabIndex        =   37
         Top             =   1680
         Visible         =   0   'False
         Width           =   255
```

```
   End
   Begin Frame Frame3
      Caption         =   "Output C---Blue Paddle"
      Height          =   975
      Left            =   4680
      TabIndex        =   12
      Top             =   120
      Width           =   2175
      Begin CommandButton Command5
         Caption         =   "Activate"
         Height          =   255
         Left            =   360
         TabIndex        =   24
         Top             =   480
         Width           =   1335
      End
   End
   Begin Frame Frame2
      Caption         =   "Input 5----Light Sensor"
      Height          =   2055
      Left            =   120
      TabIndex        =   6
      Top             =   3000
      Width           =   2175
      Begin TextBox Text4
         Enabled         =   0   'False
         Height          =   375
         Left            =   1080
         TabIndex        =   11
         Top             =   960
         Width           =   855
      End
      Begin Timer Timer2
         Enabled         =   0   'False
         Interval        =   500
         Left            =   120
         Top             =   1560
      End
      Begin TextBox Text2
         BorderStyle     =   0  'None
         Enabled         =   0   'False
         Height          =   255
         Left            =   1440
         TabIndex        =   8
         Text            =   "0.00"
         Top             =   480
         Width           =   495
      End
      Begin Label Label4
         Caption         =   "Colour"
         Height          =   255
         Left            =   120
         TabIndex        =   10
         Top             =   1080
         Width           =   975
      End
      Begin Label Label2
         Caption         =   "Light Reading"
         Height          =   255
         Left            =   120
         TabIndex        =   7
         Top             =   480
         Width           =   1335
      End
   End
```

```
   Begin Frame Frame1
      Caption         =   "Output B-White Paddle"
      Height          =   975
      Left            =   2400
      TabIndex        =   5
      Top             =   120
      Width           =   2175
      Begin CommandButton Command4
         Caption         =   "Activate"
         Height          =   255
         Left            =   360
         TabIndex        =   23
         Top             =   480
         Width           =   1335
      End
   End
   Begin CommandButton Command1
      Caption         =   "EXIT"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   13.5
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   495
      Left            =   5400
      TabIndex        =   4
      Top             =   5280
      Width           =   1215
   End
   Begin Frame OutputB
      Caption         =   "Output A------Conveyor"
      Height          =   975
      Left            =   120
      TabIndex        =   0
      Top             =   120
      Width           =   2175
      Begin OptionButton Option3
         BackColor       =   &H00FFFFFF&
         Caption         =   "On"
         FontBold        =   -1  'True
         FontItalic      =   0   'False
         FontName        =   "MS Sans Serif"
         FontSize        =   9.75
         FontStrikethru  =   0   'False
         FontUnderline   =   0   'False
         ForeColor       =   &H00000000&
         Height          =   495
         Left            =   1200
         TabIndex        =   2
         Top             =   360
         Value           =   -1  'True
         Width           =   735
      End
      Begin OptionButton Option2
         Caption         =   "Off"
         FontBold        =   -1  'True
         FontItalic      =   0   'False
         FontName        =   "MS Sans Serif"
         FontSize        =   9.75
         FontStrikethru  =   0   'False
         FontUnderline   =   0   'False
         Height          =   495
         Left            =   360
         TabIndex        =   1
```

```
         Top              =   360
         Width            =   735
      End
   End
   Begin Label Label3
      Caption         =   "Timer function"
      Height          =   495
      Left            =   7200
      TabIndex        =   9
      Top             =   1920
      Visible         =   0   'False
      Width           =   855
   End
   Begin Menu mnuFile
      Caption         =   "&File"
      Begin Menu mnuExit
         Caption         =   "&Exit"
         Shortcut        =   ^X
      End
   End
   Begin Menu mnuHelp
      Caption         =   "&Help"
      Begin Menu mnu1
         Caption         =   "-"
      End
      Begin Menu mnuAbout
         Caption         =   "About the Conveyor"
      End
      Begin Menu mnuRun
         Caption         =   "Running CONVEYOR.EXE"
      End
   End
End
```

```
Dim LightValue(8) As Integer    'Light reading refrence in Colour.Txt
Dim LightSensor As Integer      'Light sensor activator
Dim BlockColour(8) As String    'Light colour refrence in Colour.Txt
Dim NumEntries As Integer
Dim LightRawValue(10) As Integer  'Raw data from light sensor
Dim RawConvValue(10) As Integer
Dim NumBlockColour As Integer
Dim LAvg As Integer
Dim BlueTable(1 To 10, 1 To 2)    'Blue pusher activator
Dim WhiteTable(1 To 10, 1 To 2)  'White pusher activator
Dim CountT As Integer
Dim AutoOn As Integer
Dim TableInsertActiv As Integer  'Activates array insertion
Dim j As Integer

Const OK = 0
Const WrongText = 1
Const WrongPort = 2
Const NoInterfaceBoxPresent = 3

Declare Function IFB_InitLegoInterface Lib "lego.dll" Alias "#31" (ByVal init
text As String, ByVal Answertext As String, ByVal portnr As Integer) As Integ
er

Declare Function IFB_Raw Lib "lego.dll" Alias "#30" (ByVal InputPort As Integ
er) As Integer

Declare Function IFB_Temperature Lib "lego.dll" Alias "#29" (ByVal InputPort
As Integer) As Integer

Declare Function IFB_Boolean Lib "lego.dll" Alias "#28" (ByVal InputPort As I
nteger) As Integer

Declare Function IFB_Light Lib "lego.dll" Alias "#27" (ByVal InputPort As Int
eger) As Integer

Declare Function IFB_Count Lib "lego.dll" Alias "#26" (ByVal InputPort As Int
eger) As Integer

Declare Function IFB_Steps Lib "lego.dll" Alias "#25" (ByVal InputPort As Int
eger) As Integer

Declare Sub IFB_JitterfilterOn Lib "lego.dll" Alias "#24" (ByVal InputPort As
Integer)

Declare Sub IFB_JitterfilterOff Lib "lego.dll" Alias "#23" (ByVal InputPort A
s Integer)

Declare Sub IFB_SetCelsius Lib "lego.dll" Alias "#22" ()

Declare Sub IFB_SetFahrenheit Lib "lego.dll" Alias "#21" ()

Declare Sub IFB_SetDynamicDigital Lib "lego.dll" Alias "#20" (ByVal InputPort
As Integer, ByVal DynamicValue As Integer)

Declare Sub IFB_FastsamplingOn Lib "lego.dll" Alias "#19" (ByVal ListOfTwoInp
utPorts As String)

Declare Sub IFB_FastsamplingOff Lib "lego.dll" Alias "#18" ()

Declare Sub IFB_OnLeft Lib "lego.dll" Alias "#17" (ByVal OutputPortList As St
ring)

Declare Sub IFB_OnRight Lib "lego.dll" Alias "#16" (ByVal OutputPortList As S
tring)
```

```
Declare Sub IFB_On Lib "lego.dll" Alias "#15" (ByVal OutputPortList As Strin

Declare Sub IFB_ReverseDirection Lib "lego.dll" Alias "#14" (ByVal OutputPor
List As String)

Declare Sub IFB_Off Lib "lego.dll" Alias "#13" (ByVal OutputPortList As Stri
g)

Declare Sub IFB_OffFloat Lib "lego.dll" Alias "#12" (ByVal OutputPortList As
String)

Declare Sub IFB_SetLeft Lib "lego.dll" Alias "#11" (ByVal OutputPortList As
tring)

Declare Sub IFB_SetRight Lib "lego.dll" Alias "#10" (ByVal OutputPortList As
String)

Declare Sub IFB_FlashOn Lib "lego.dll" Alias "#9" (ByVal OutputPortList As S
ring, ByVal OnTime As Integer, ByVal OffTime As Integer)

Declare Sub IFB_FlashOff Lib "lego.dll" Alias "#8" (ByVal OutputPortList As
tring)

Declare Sub IFB_SetPower Lib "lego.dll" Alias "#7" (ByVal OutputProtList As
tring, ByVal Levels As Integer)

Declare Sub IFB_OnFor Lib "lego.dll" Alias "#6" (ByVal OutputPortList As Str
ng, ByVal OnforTime As Integer)

Declare Function IFB_Stoped Lib "lego.dll" Alias "#5" () As Integer

Declare Sub IFB_ResetInputPort Lib "lego.dll" Alias "#32" (ByVal InputPort A
Integer)

Declare Function IFB_Reinitialized Lib "lego.dll" Alias "#4" () As Integer

Declare Sub IFB_AdjustTemperature Lib "lego.dll" Alias "#3" (ByVal InputPort
As Integer, ByVal Degree As Integer)

Declare Sub IFB_AdjustLight Lib "lego.dll" Alias "#2" (ByVal InputPort As Int
eger, ByVal Light As Integer)

Sub Option1_Click ()
    IFB_OnRight "a"
End Sub

Sub Command1_Click ()
    End
End Sub

Sub Command10_Click ()
    AutoOn = 0 'stops automatic function
    Command3.Visible = True
End Sub

Sub Command2_Click ()
    If HScroll2.Value = 0 Then
     HScroll2.Value = 1
    End If
    AutoOn = 1 'Set to Automatic opperation
End Sub
```

```
Sub Command3_Click ()
    Timer3.Enabled = True
    Command7.Visible = True
End Sub

Sub Command4_Click ()
    IFB_OnFor "b", 5
End Sub

Sub Command5_Click ()
    IFB_OnFor "c", 5
End Sub

Sub Command6_Click ()
    AutoOn = 2 'Set to Manual opperation
    Timer2.Enabled = True
End Sub

Sub Command7_Click ()
    Frame6.Visible = True
    Command3.Visible = False
End Sub

Sub Command8_Click ()
    Frame4.Visible = True
    Command2.Visible = False
    Command6.Visible = True
    Frame6.Visible = False
    Command7.Visible = False
    Command10.Visible = True
End Sub

Sub Command9_Click ()
    Command2.Visible = True
    Command6.Visible = False
    Frame1.Enabled = False
    Frame3.Enabled = False
    Frame4.Visible = True
    Frame6.Visible = False
    Command7.Visible = False
    Command4.Enabled = False
    Command5.Enabled = False
    Label9.Visible = True
    HScroll2.Visible = True
    Text11.Visible = True
    Command10.Visible = True
End Sub

Sub Form_Load ()
Dim i As Integer
    i = IFB_InitLegoInterface("Do you byte, when I knock?", "Just a bit off t
he block!", 2)
    If i = WrongText Then
       MsgBox "Wrong text", MB_OK, "Error during integerface B initializatior
"
       End
    ElseIf i = WrongPort Then
       MsgBox "Wrong port", MB_OK, "Error during integerface B initializatior
"
       End
    ElseIf i = NoInterfaceBoxPresent Then
       MsgBox "No interfacebox present", MB_OK, "Error during integerface B i
nitialization"
       End
    End If
```

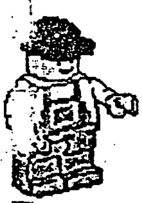
```
    AutoOn = 0
    ' Read data from disc file into array
    Open "c:\data\vb\thesis\colour.txt" For Input As #9 'c:\data\vb\thesis\
lour.txt
    NumEntries = 0
        Do
            NumEntries = NumEntries + 1
            Input #9, LightValue(NumEntries), BlockColour(NumEntries)
        Loop Until EOF(9)
   'procedure to clear pusher tables
    CountT = 0
    For CountT = 1 To 10
        BlueTable(CountT, 1) = 0    'Initialises the table
        WhiteTable(CountT, 1) = 0   'Initialises the table
    Next CountT
End Sub

Sub HScroll2_Change ()

    Timer2.Interval = 60 / HScroll2.Value * 1000
        'Opperations per second * 1000 to configure Timer4
    Text11.Text = HScroll2.Value
End Sub

Sub mnuAbout_Click ()
    'Open helpabou.frm
    HelpAbout.Show 1
End Sub

Sub mnuExit_Click ()
    End
End Sub

Sub mnuRun_Click ()
    'Open helpRun.frm
    HelpRun.Show 1

End Sub

Sub Option2_Click ()
    IFB_Off "a"
End Sub

Sub Option3_Click ()
   IFB_OnLeft "a"
   IFB_SetPower "a", 4
End Sub

Sub Option4_Click ()
    IFB_On "d"
End Sub

Sub Option5_Click ()
    IFB_Off "d"
End Sub

Sub Option6_Click ()
    IFB_On "e"
End Sub

Sub Option7_Click ()
    IFB_Off "e"
End Sub

Sub Text4_Change ()
```

```
    'Activate pusher alert
    PneuB = 0
    PneuW = 0
    Select Case Text4.Text
        Case "White"
            PneuW = Val(Text6.Text) + 51

        Case "Blue"
            PneuB = Val(Text6.Text) + 19
        End Select
End Sub

Sub Timer1_Timer ()
     CountT = 1
    'Monitor for conveyor distance
    Text6.Text = Format(IFB_Steps(7), "######")
        Text10.Text = IFB_Boolean(1) 'Checks Touch Sensor
    'checks the array to see if a block is in place to be pushed off
    If BlueTable(1, 1) = False Then  'No blocks in Array
        End If

    If BlueTable(1, 2) = Val(Text6.Text) And BlueTable(1, 1) = 1 Then
        IFB_OnFor "c", 5
        BlueTable(1, 1) = 0 'Reinitialise the array
    Do                        'Reorganize the Array
        BlueTable(CountT, 1) = BlueTable(CountT + 1, 1)
        BlueTable(CountT, 2) = BlueTable(CountT + 1, 2)

        If BlueTable(CountT + 2, 1) = 0 Then
            BlueTable(CountT + 1, 1) = 0
            Exit Sub
        Else
            CountT = CountT + 1
        End If
    Loop
    End If
    If WhiteTable(1, 1) = False Then
        End If
    If WhiteTable(1, 2) = Val(Text6.Text) And WhiteTable(1, 1) = 1 Then
        IFB_OnFor "b", 5
        WhiteTable(1, 1) = 0 'Reinitialise the array
    Do
        WhiteTable(CountT, 1) = WhiteTable(CountT + 1, 1)
        WhiteTable(CountT, 2) = WhiteTable(CountT + 1, 2)
        If WhiteTable(CountT + 2, 1) = 0 Then
            WhiteTable(CountT + 1, 1) = 0
            Exit Sub
        Else
            CountT = CountT + 1
        End If
    Loop
    End If
End Sub

Sub Timer2_Timer ()
    'Procedure to activate feeder
    If AutoOn > 0 Then      'Activates feeder mechanism

       'activate pneumatic magazine feeder
       Option4.Value = True     'activate pusher
       For j = 1 To 500
        Text18.Text = j
       Next j
        'checks for block in magazine
        If IFB_Boolean(1) = 1 Then    'Block in magazine
```

```
            'Identifies the blocks colour
            Dim K As Integer
            Dim KMax As Integer
            K = 0
            KMax = 0
            For K = 1 To 10
                LightRawValue(K) = Format(IFB_Light(5) / 100#, "#00")
                    If LightRawValue(K) > KMax Then
                        KMax = LightRawValue(K)
                    End If
            Next K
            NumBlockColour = (KMax - LAvg)
            LightSensor = 1
            TableInsertActiv = 1
           'Colour identified, activate pusher

              Option6.Value = True 'Activate pusher
              For j = 1 To 200
                Text18.Text = j
              Next j
              Option4.Value = False
              Option5.Value = True
              Option6.Value = False
              Option7.Value = True
        End If


        If IFB_Boolean(1) = 0 Then         'No Block in magazine
           Option4.Value = False
           Option5.Value = True
        End If

        If AutoOn = 2 Then ' Resets manual opperation, only allows one operat
ion

           AutoOn = 0
        End If


        IFB_Off "e"
       End If


    Text4.Enabled = True
    text2.Text = Format(IFB_Light(5) / 100#, "#00")
    Text8.Text = NumBlockColour

    'Identifies colour
    If LightSensor = 1 Then
        Dim Message As String
        Dim N As Integer
        Message = "Links"
        For N = 1 To NumEntries
            If Val(Text8.Text) = LightValue(N) Then
                Message = BlockColour(N)
            End If
        Next N
        Text4.Text = Message
        LightSensor = 0
    End If

        'Insert colour into lookup Table
      If TableInsertActiv = 1 Then     'Checks whether to insert into table
        Select Case Message
            Case "Blue"
```

```
                CountT = 1
                For CountT = 1 To 10
                    If BlueTable(CountT, 1) = 0 Then
                        BlueTable(CountT, 1) = 1     'Puts Blue block into t
ble
                        BlueTable(CountT, 2) = Val(Text6.Text) + 23 'Sets Pu
he
                        TableInsertActiv = 0
                        Exit For
                    End If
                Next CountT

            Case "White"   'Call Insert White
                CountT = 1
                For CountT = 1 To 10
                    If WhiteTable(CountT, 1) = 0 Then
                        WhiteTable(CountT, 1) = 1     'Puts White block into
table
                        WhiteTable(CountT, 2) = Val(Text6.Text) + 57 'Sets Pu
sher
                        TableInsertActiv = 0
                        Exit For
                    End If
                Next CountT

            Case "Links"

        End Select
       End If
    Timer2.Enabled = False
End Sub

Sub Timer3_Timer ()
        Dim L As Integer
        L = 0
        LAvg = 0
        For L = 1 To 10
            RawConvValue(L) = Format(IFB_Light(5) / 100#, "#00")
            LAvg = LAvg + RawConvValue(L)
        Next L
        LAvg = LAvg / 10
        text7.Text = LAvg
        Timer3.Enabled = False
End Sub
```

Conveyor Version 1.0



This programe was developed by Ari Levin
Department of Mechanical Engineering
University of Cape Town, South Africa.

For the controlling of a LEGO
model of an Industrial sorting
system.

OK

```
VERSION 2.00
Begin Form HelpAbout
   Caption         =   "About Conveyor"
   ClientHeight    =   2595
   ClientLeft      =   2355
   ClientTop       =   1470
   ClientWidth     =   5010
   Height          =   3000
   Left            =   2295
   LinkTopic       =   "Form2"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   2595
   ScaleWidth      =   5010
   Top             =   1125
   Width           =   5130
   Begin CommandButton Command1
      Caption         =   "OK"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   9.75
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   375
      Left            =   3840
      TabIndex        =   0
      Top             =   1920
      Width           =   855
   End
   Begin Image Image1
      Height          =   1350
      Left            =   120
      Picture         =   (Bitmap)
      Top             =   480
      Width           =   1020
   End
   Begin Label Label3
      Caption         =   "For the controlling of a LEGO model of an Industri
      Height          =   735
      Left            =   1200
      TabIndex        =   3
      Top             =   1560
      Width           =   3015
   End
   Begin Label Label2
      Caption         =   "This programe was developed by Ari Levin Departmen
      Height          =   735
      Left            =   1200
      TabIndex        =   2
      Top             =   720
      Width           =   3615
      WordWrap        =   -1  'True
   End
   Begin Label Label1
      Caption         =   "Conveyor Version 1.0"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   9.75
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   375
```

```
      Top             =    120
      Width           =    2295
      WordWrap        =    -1   'True
   End
End
```

```
Sub Command1_Click ()
    Unload HelpAbout
End Sub
```

Conveyor Version 1.0

This programme was designed to run on an AcerMate 433s 486 SX 33 PC running this software on a slower or faster machine could cause problems with the synchronizing of the model.

OK

```
VERSION 2.00
Begin Form HelpRun
   Caption         =   "Running CONVEYOR.EXE"
   ClientHeight    =   2640
   ClientLeft      =   2310
   ClientTop       =   4575
   ClientWidth     =   5010
   Height          =   3045
   Left            =   2250
   LinkTopic       =   "Form2"
   ScaleHeight     =   2640
   ScaleWidth      =   5010
   Top             =   4230
   Width           =   5130
   Begin CommandButton Command1
      Caption         =   "OK"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   9.75
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   375
      Left            =   3840
      TabIndex        =   1
      Top             =   1920
      Width           =   855
   End
   Begin Label Label2
      Caption         =   "Conveyor Version 1.0"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   9.75
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   375
      Left            =   1200
      TabIndex        =   2
      Top             =   120
      Width           =   2295
      WordWrap        =   -1  'True
   End
   Begin Label Label1
      Caption         =   "This programme was designed to run on an AcerMate
      Height          =   1215
      Left            =   1560
      TabIndex        =   0
      Top             =   600
      Width           =   3135
   End
   Begin Image Image1
      Height          =   1350
      Left            =   240
      Picture         =   (Bitmap)
      Top             =   360
      Width           =   1020
   End
End
```

HELPRUN.FRM - 1

```
Sub Command1_Click ()
    Unload HelpRun
End Sub
```

Appendix 8 CONVEYOR.EXE Manual

CONVEYOR HELP
Version 1.0

This programme was written by Ari Levin of the Department of Mechanical Engineering, University of Cape Town. As part of a Masters in Industrial Administration.

The files contained for this programme are:

CONVEYOR.EXE	The main programme
COLOUR.TXT	Text file with the light sensor information
HELP.WRI	This File
NUKE.ICO	Icon
VBRUN300.DLL	Visual Basic DLL file
*LEGO.DLL	Lego interface DLL file

LEGO.DLL file is copyrighted and may not be distributed without the permission of Lego AG

This programme in conjunction with a Lego model of a conveyor belt. The programme is written in Microsoft Visual Basic 3.0 and will operate through Windows version 3.x. The conveyor sorts Lego flat 2 by 2 blocks according to their colour. This programme is designed to sort blue and white colours only.

Installing the Program

Copy all the files into a directory called Conveyor. The DLL files can be copied into the Windows directory or the conveyor directory should be added into the path.

Running the Program

The options that can be manipulated in this programme are the
Conveyor belt speed
The frequency of feeding the blocks

Feeder Mechanism

The Button labelled "Feeder Control" will activate a new window for Manual or Automatic control.

Manual Control

Selecting Manual control will allow the user to control the feeding of blocks onto the conveyor belt

Automatic Control

Selecting Automatic control will show a new window "Output D E---Feeder" in this window is a button "Auto Activate" and a scroll bar

Scroll bar

The scroll bar will set the frequency at which the blocks will be fed onto the conveyor belt.

Auto Activate

This will start the feeder mechanism working at the rate set by the scroll bar.

Stop

This will stop the feeding mechanism when set on automatic

Conveyor speed

The conveyor speed can be Changed between 0 and 8. This is done through the scroll bat in the frame labelled Output A-----Conveyor. The speed or rate of the conveyor belt can be seen in the window labelled "Input 6 7-----Angular" under "Conveyor Counter".

Light Sensor

In order for the light sensor to work a reference light reading must be taken. This can be done several times as the ambient light changes. MAKE sure there are no blocks in under the light sensor while doing this operation.

The light reading from the light sensor can be seen in window "Input 5---Light Sensor" The figure opposite "Light Reading is the raw light reading from the sensor. The colour of the block will be indicated in the window next to colour.

Connecting the Model

Connect the Lego interface B into serial port 2.
If there is an error "Can't find control box" then connect interface to serial port 2

Connecting the electrical leads

From the model

Output A	Conveyor motor
Output B	To relay 3
Output C	To relay 4
Output D	To relay 1
Output E	To relay 2

Input 1 (Yellow)	To touch sensor on feeder
Input 5 (Blue)	To light sensor on feeder
Input 7 (Blue)	To angle sensor on conveyor belt

From the relays

Relays read from left to right, with transformer behind the relays.
Valves read from left to right with pneumatic hose connector on the right

Relay 1	To valve 2
Relay 2	To valve 1
Relay 3	To valve 4
Relay 4	To valve 3

Appendix 9 Courses completed for this degree

Quantitative Methods	(5 Credits)
Introduction to Business	(5 Credits)
Project Management	(5 Credits)
Operations Management	(5 Credits)