UNIVERSITY OF TORONTO

INSTITUTE FOR AEROSPACE STUDIES

4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6

THE WOOD PACKER

"HOW MUCH WOOD WOULD A WOOD PACKER PACK IF A WOOD PACKER WOULD PACK WOOD?"

prepared by

Team 59 – Wednesday

Sherry Shi (999064033)
Arushri Swarup (998866071)
Andrew Talor Wong (999066816)

prepared for

Prof. M.R. Emami

A technical report submitted for
AER201 – Engineering Design

TA: Michael Czajka

April 15, 2013

TEAM PHOTO

ACKNOWLEDGMENTS

The members of Team 59, Sherry Shi, Arushri Swarup, and Andrew Wong, would like to acknowledge the help of the course instructor, Professor Reza Emami, and their teaching assistant, Michael Czajka for their assistance throughout the semester. The lab engineer, Charles Lin is also acknowledged for providing assistance in the electromechanical lab.

ABSTRACT

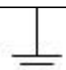
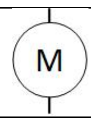
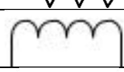
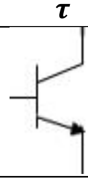
This project addresses the Request for Proposal (RFP) from a utility company. The company needs to package two colours of wooden dowels in boxes in various patterns specified by the user. The main design objectives are user friendliness, physical robustness and ease of manufacturing and maintenance.

The solution is composed of three subsystems: electromechanical, circuits and microcontroller. Before the operation begins, the user must load the dowels and the boxes into their corresponding reservoirs. The user then selects two of six available packaging patterns through the LCD-keypad user interface. As the operation starts, a box will be loaded onto the conveyor belt. As the box moves across the conveyor belt, the machine will successively drop the wooden dowels into the slots in the box. After the box is completely packaged, the lid will close as the box moves through a closing mechanism, at which point it is ready for pickup. The cost for this design is $142.02.

Full integration of the system was not successful due to unpredicted problems, mainly involving the integration of the circuits and the microcontroller, as well as limitations due to relying on gravity to align incoming dowels. Nonetheless, some of the successes of this design include its simple and modular components, cost efficiency and the lightweight yet robust structure.

TABLE OF CONTENTS

NOTATIONS AND SYMBOLS

Notation or Symbol	Definition
$\boldsymbol{a}$	Acceleration. Measured in m/s^2
A	Amperes. Unit of measurement of current
	Capacitor Circuit Symbol
$\boldsymbol{I}$	Current (I)
$\boldsymbol{d_l}$	Diameter of larger gear (m)
$\boldsymbol{d_m}$	Diameter of smaller gear (m)
	Diode Circuit Symbol
F	Farad. Unit of measurement of capacitance
$\boldsymbol{F}$	Force
$\boldsymbol{\eta}$	Gear Ratio
G	Gram. Unit of measurement of mass
	Ground (0V reference) Circuit Symbol
Hz	Hertz. Unit of measurement of frequency, also s^{-1}
$\boldsymbol{m}$	Mass, measured in kg
	Motor Circuit Symbol
N	Newton. Measure of force
Ω	Ohm. Unit of measurement of resistance
$\boldsymbol{R}$	Resistance (Ω)
R	Resistance (Ω)
	Resistor Circuit Symbol
	Solenoid (Inductor) Circuit Symbol
	Switch Circuit Symbol
$\boldsymbol{\tau}$	Torque (N·m)
	Transistor (NPN) Circuit Symbol
$\boldsymbol{I_B}$	Transistor Base Current (A)
$\boldsymbol{I_C}$	Transistor Collector Current (A)
$\boldsymbol{H_{FE}}$	Transistor Forward Current Gain
$\boldsymbol{V}$	Voltage (V)
V	Volts. Unit of measurement of electric potential
W	Watt. Unit of measurement of power

Abbreviations

Abbreviation	Definition
AHP	Analytic Hierarchy Process
AWG	American Wire Gauge
DC	Direct Current
EEPROM	Electronically Erasable Programmable Read-Only Memory
EM	Electromagnetic
GND	Ground (0V reference)
I/O Ports	Input/Output Ports
ISR	Interrupt Service Routine
KISS	Keep it Simple, Stupid
LCD	Liquid Crystal Display
PIC	Peripheral Interface Controller
PWM	Pulse Width Modulation
RFP	Request for Proposals
RTC	Real Time Clock

1. INTRODUCTION

Today, much of the manufacturing industry is adopting automated processes. The advantages of automation include lower costs, better efficiency and higher reliability [1]. Historically, industrial machines are known for their size and power. In modern society however, more compact and energy efficient designs are preferred [2]. Nonetheless, features such as speed, reliability, user-friendliness and cost efficiency are still some of the most valuable aspects in many designs. The request for proposal [3] which this technical report addresses, requests for a dowel packaging machine which can package two types of colored dowels in set patterns.

The machine designed by Team 59 aimed to use the most simplistic methods to achieve its task, thereby reducing complexity of the system. However, this machine was not successful mainly due to unforeseen problems encountered during the integration of the circuits and the microcontroller. The modular approach to this design allowed us to pinpoint the root of the problem to be a malfunctioning power supply. This report explains the designed machine in detail, breaking the entire machine into three subsystems: electromechanical, circuits, and microcontroller.

2. PERSPECTIVE

2.1 THEORY AND HISTORY

Use of Gravity

As our design objectives called for simplicity in manufacturing, operation and maintenance, the design relies on the simple physics of gravity to perform the tasks of sliding the boxes into place on the conveyor belt and aligning the dowels. The design objective for loading the dowels into the machine is stated in the RFP as “the dowels are to be supplied in no particular format into two separate reservoirs in the machine” [3]. Therefore, the loading mechanism was to be designed to accommodate for the worst possible way to load dowels into the reservoir which would be to load them all in at once, or throw them in. The limitations encountered were due to the unpredictable randomness of dowel arrangement when dropped from a height.

Gravity was relied upon to load the dowels into the box as well. As the dowels fell down into the reservoir, they rolled down and into position for ejection.

Linear force, done by a solenoid, was utilized to push dowels out of the reservoir to be loaded. The force exerted by the solenoid needed to overcome the force of gravity of the dowel in order to push it out.

Gravity also helped the box slide into place on the conveyor mechanism after exiting the box loader. The boxes were loaded onto the box loader’s slanted shelves that allowed the box to slide down out of its shelf when it was lifted above the conveyor onto the loading ramp.

$F = mg$, the force of gravity, was a force used widely throughout the machine. The KISS rule, "Keep It Simple, Stupid," was employed throughout the design. Thus the design relied on gravity, as it is the force that never fails; "what goes up must come down," as Isaac Newton said [4]. Keeping the design simple deems it a reliable one. A design requiring many actuators has many points of failure, and it makes the probability of the whole machine working as a whole decrease compared to a design which uses minimal actuators. This is the reason for utilizing gravity so widely in the design.

DC Motors

Another design decision was the use of a DC motor to drive the box-loading elevator to lift the boxes. In order to lift the shelves containing at most three empty boxes, the motor-gearbox mechanism needed to produce enough torque to perform the task. The goal was to use a simple motor in terms of circuitry and performance: a 12V DC brush motor. The limitations of these motors are due to their typically high speed, low torque. This can be overcome using a gear system to amplify torque and decrease speed. Torque, $\tau = FR$, was amplified by a spur gear system of ratio: $\eta = \frac{d_l}{d_m} = \frac{\tau_o}{\tau_i}$.

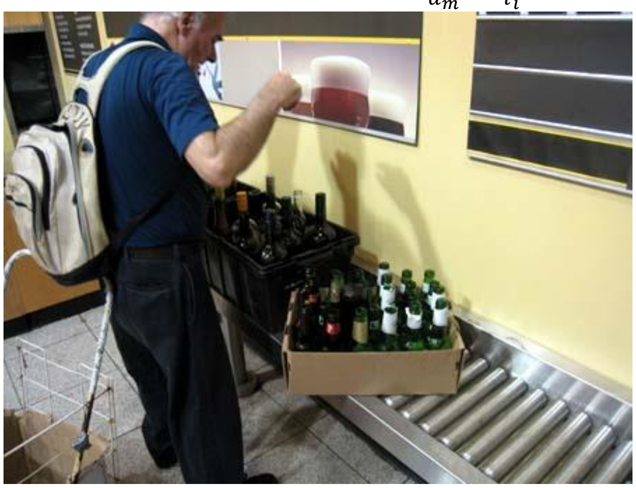


Figure 2-1: Rolling pins are used to support the boxes as they move along the conveyor. *[6]*

Figure 2-2: Candy Dispenser mechanism *[7]*

2.2 SURVEY

Conveyer Belt

One idea that has been consistent since the start of the project was to move the box along a conveyer belt as it goes through the operation. This method requires only linear, unidirectional motion, which is the simplest to implement. One of the concerns when designing the conveyor belt is the slack that may result from the flexible belt material. In the following photo, rolling pins along the path are a simple way to keep the boxes levelled and allow the box to travel smoothly. This was initially adopted by the proposed design. However, it was later replaced by a smooth piece of hardboard under the conveyer belt, which is even simpler.

Dowel Dispensing

One of the more complex stages is the dispensing of the dowels. One of the challenges is that the dowels must be dispensed one at a time. The following are some existing methods that allow unit-wise dispensing: candy dispenser, and straw dispenser, as shown in Figures 2-2 and 2-3. A

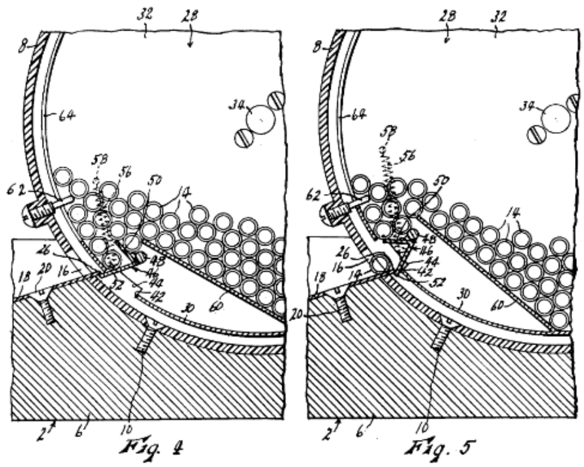


Figure 2-3: Candy Dispenser schematic. *[8]*

commonality in the two designs is that the items are loaded and ready to be dispensed when the machine is on standby. The mechanism also blocks the items in the reservoir from leaving when the unit is dispensed.

Reservoir

This design is used by the Keewa Industries Corporation [3]. It consists of a large inverted triangular prism shaped reservoir with pins attached to the walls which, when avalanched by straws, aids in their horizontal alignment. This design was tested in the lab, by horizontally attaching pencils inside a cardboard box and dropping dowels into the reservoir. The design was unsuccessful in aligning the dowels horizontally. As such, a different design was needed and implemented in the final product.

3. OBJECTIVES

3.1 INTRODUCTION

This technical report addresses the Request for Proposal (RFP) from a utility company [4]. The company needed to package two colors of wooden dowels in organizer boxes in various mixed combinations. The machine is expected to package 2 boxes of 6 dowels each in under 2 minutes. It is expected to hold at least 3 boxes and at most 12 dowels of each color before the operation commences. The method of loading the boxes and dowels are up to the design, but must be convenient for the operator. In addition, the dowels are to be loaded in no particular order. In each box, 6 dowels of a particular pattern are to be packed as specified by the operator through a keypad before the operation. After loading each box, the lid must be snapped shut and moved to its designated pickup area. After each operation, the machine is expected to return to standby mode to await further commands. The client requires that the machine be portable with no need for installations, and as such there are constraints on weight and dimensions. Also for safety purposes the machine must have an easily-accessible emergency STOP switch that stops all the mechanical moving parts immediately. The machine can be plugged into the AC outlet.

The full specifications outlined in the Request for Proposal, which is found in Appendix D.

3.2 CONSTRAINTS

The machine must be designed with the following constraints, adapted from the request for proposals [4]. See Appendix D for the full list of constraints.

a. The entire prototype shall completely fit within a 75x75x75 cm^3 envelope at all operation times.
b. The weight of the machine shall not exceed 6 kg.

c. The total prototype costs shall not exceed $230CDN.
d. The machine must have an easily-accessible emergency STOP switch that stops all the mechanical moving parts immediately.
e. The machine must be fully autonomous, and no interaction with an external PC or remote control is permitted during the operation. The operation must start by pressing a <start> button on the keypad.
f. No installation or instrumentation is allowed in addition to what is devised within the machine. Dispensing the packed boxes and retrieving the returning dowels must be easy with no need for disassembling any parts of the machine.
g. The locations for supplying dowels and boxes and also the pickup location/bin must be specified in the machine clearly.
h. At the end of each run, the machine display must be on prompt to show the following information per operator’s request: operation time, number of dowels remaining in each reservoir, and the packaging pattern for each box.
i. The machine user interface for both operation and information retrieval shall be self-explanatory, and provide easy navigation for operators of various skill levels.
j. The operation time shall not exceed 2 minutes. Further, the time required for entering the operator’s instructions on the keypad before the operation shall not exceed 1 minute.
k. The machine must pose no hazard to the operator, and shall not be perceived as hazardous (e.g., too much vibration or noise or frequent sparks during the operation is perceived as dangerous.)

3.3 Acceptance Criteria in Decision Making

The design decisions were made with the design requirements and constraints in mind. The acceptance criteria grouped into three main categories - user-friendliness, physical properties and manufacturability.

User-friendliness

- Operability - Can this component be easily integrated with the rest of the system? Will the component allow the system to operate efficiently and effectively?
- Safety - Does the component put the user at risk of any physical harm? Will it evoke a fear of danger in the user?
- Reliability - The success rate of the component under various operation circumstances and environments (e.g. will it succeed even when other components fail?)
- Cost - Monetary value of the materials.

Physical Properties

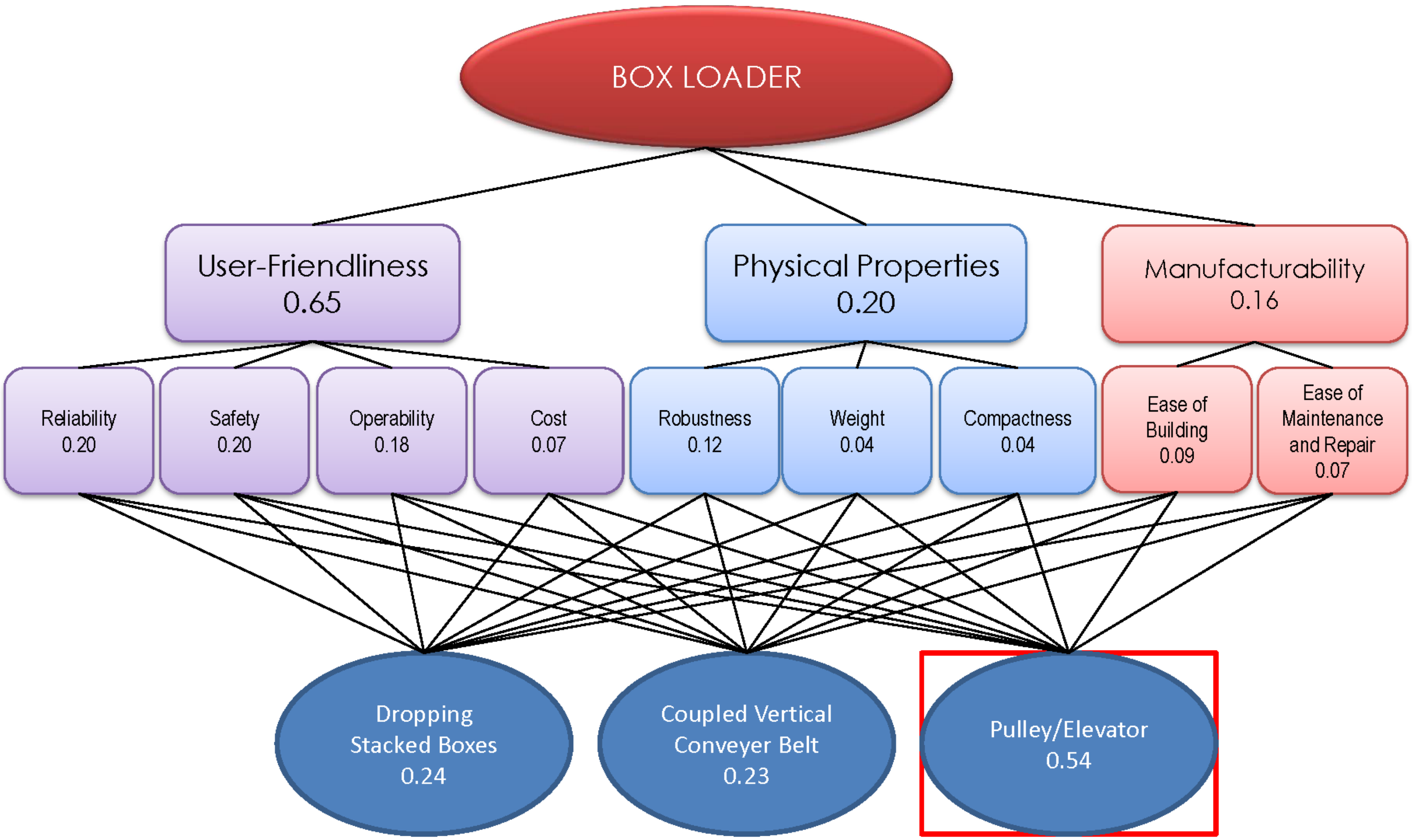
- Robustness - Strength of the material; can it be integrated with the rest of the system in a secure manner? How well the component can resist damage.
- Weight - A body's relative mass or the quantity of matter contained by it (the maximum weight of the entire system is 6 kg).

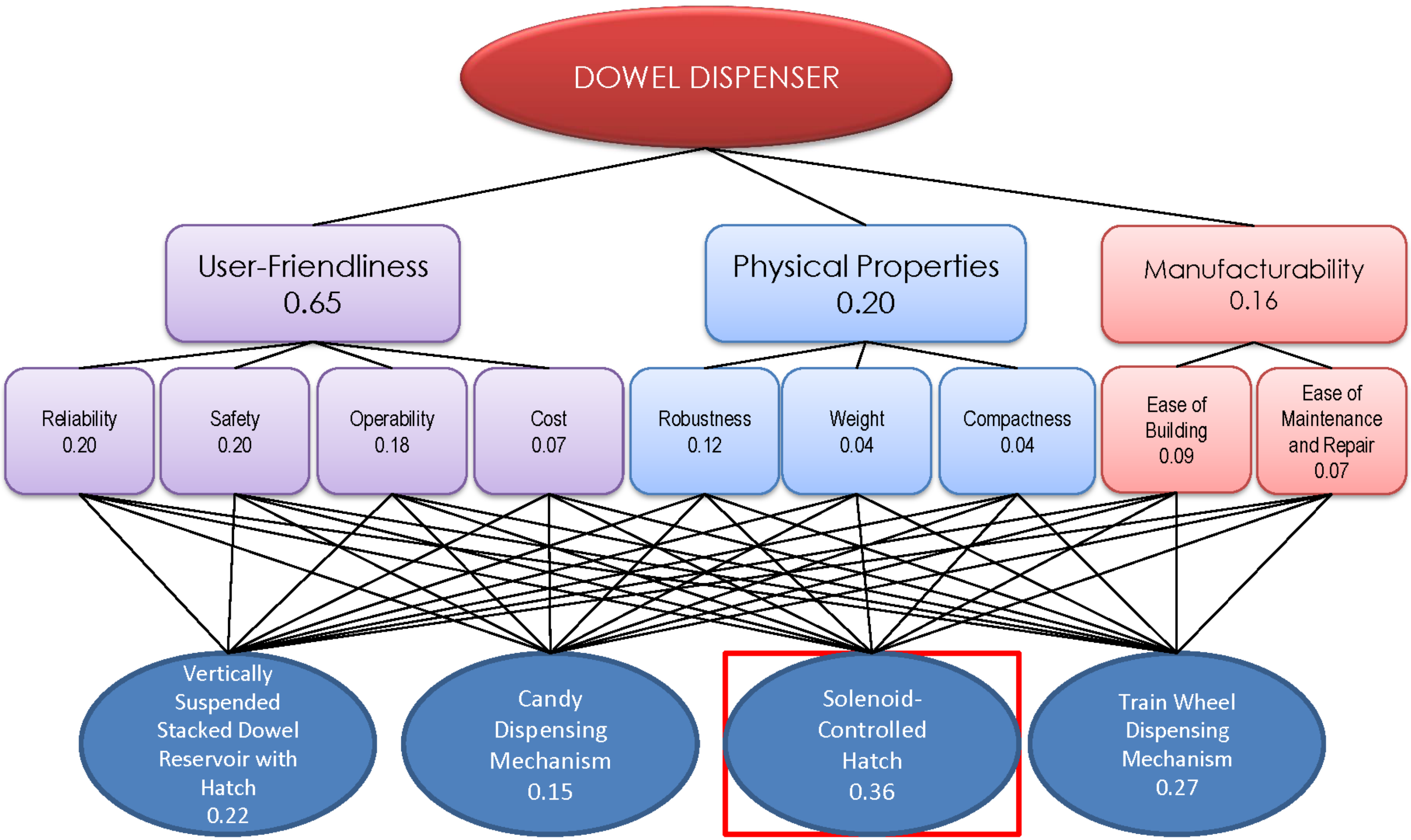
- Compactness - the amount of space required; the efficiency of the use of space of the component itself and the allowance of efficient use of space of the rest of the system

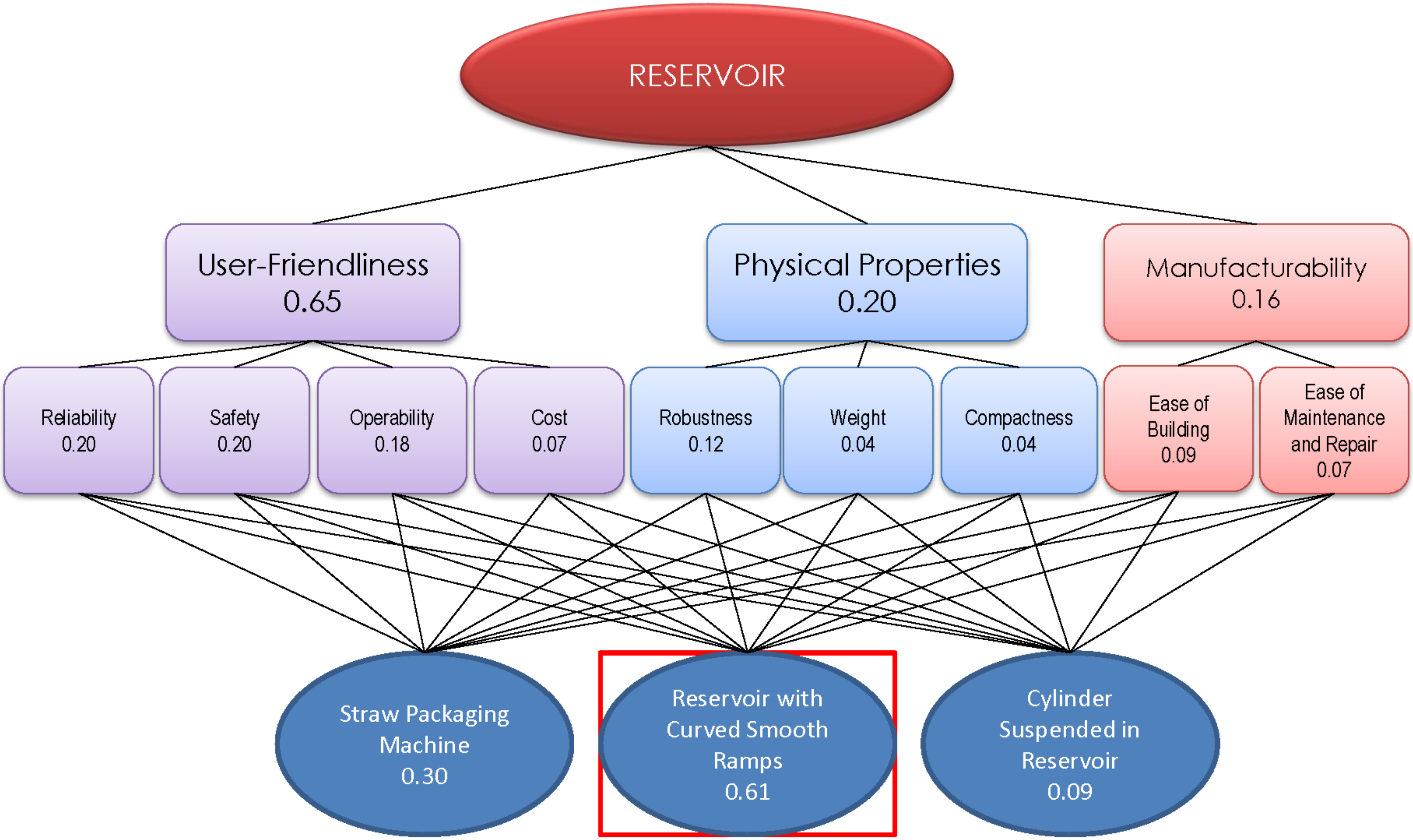
Manufacturability

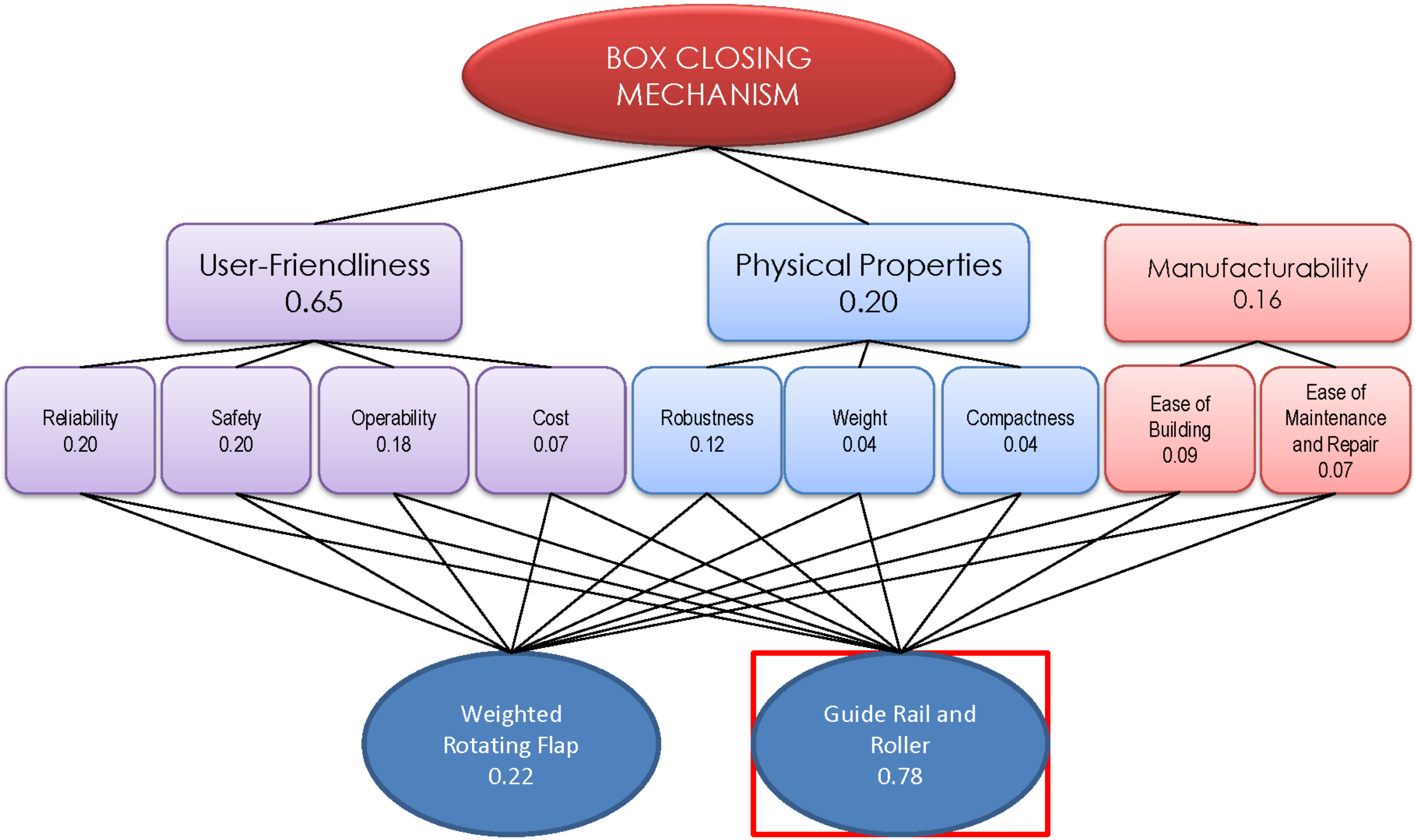
- Ease of Building - time and effort required to build the component
- Ease of Maintenance and Repair - time, effort required to maintain and repair the component, and frequency of repair required.

3.4 AHP ANALYSIS




DOWEL DISPENSER
User-Friendliness 0.65
Physical Properties 0.20
Manufacturability 0.16
Reliability 0.20
Safety 0.20
Operability 0.18
Cost 0.07
Robustness 0.12
Weight 0.04
Compactness 0.04
Ease of Building 0.09
Ease of Maintenance and Repair 0.07
Vertically Suspended Stacked Dowel Reservoir with Hatch 0.22
Candy Dispensing Mechanism 0.15
Solenoid-Controlled Hatch 0.36
Train Wheel Dispensing Mechanism 0.27


RESERVOIR
User-Friendliness
0.65
Physical Properties
0.20
Manufacturability
0.16
Reliability
0.20
Safety
0.20
Operability
0.18
Cost
0.07
Robustness
0.12
Weight
0.04
Compactness
0.04
Ease of Building
0.09
Ease of Maintenance and Repair
0.07
Straw Packaging Machine
0.30
Reservoir with Curved Smooth Ramps
0.61
Cylinder Suspended in Reservoir
0.09


BOX CLOSING MECHANISM
User-Friendliness 0.65
Physical Properties 0.20
Manufacturability 0.16
Reliability 0.20
Safety 0.20
Operability 0.18
Cost 0.07
Robustness 0.12
Weight 0.04
Compactness 0.04
Ease of Building 0.09
Ease of Maintenance and Repair 0.07
Weighted Rotating Flap 0.22
Guide Rail and Roller 0.78

3.5 BUDGET

Budget	Part:	Material:	Dimensions (cm or cm2) or Quantity	Cost($)/ Unit (cm2 or cm)	Part Cost ($):	Supplier:
Section						
Box Loader	Actuator	Motor	1	5	5.00	Design Store
	Gears	Plastic Gears	2	0.5	1.00	Active Surplus
	Frame	2X1 Framing Lumber	255	0.00422	1.08	Home Depot
	Shelves	Hardboard	1793.5	0.00029	0.52	Home Depot
	Slider Guide	Corrugated Plastic	108	0.0014	0.15	Home Depot
	String for Pulley	String	100	0.0005	0.05	Home Depot
	Shaft for pulley	Bolt	1	2	2.00	Home Hardware
		Nut	5	0.1	0.50	Home Hardware
	Guide for Boxes	Hardboard	400	0.00029	0.12	Home Depot
	Ramp	Hardboard	187	0.00029	0.05	Home Depot
	Supporter of Ramp	2X1 Framing Lumber	11.5	0.00422	0.05	Home Depot
Dowel Loader	Actuator	Solenoid	2	5	10.00	Design Store
	Frame (beneath the loader)	2X1 Framing Lumber	80	0.00422	0.34	Home Depot

Aligner	Dowel Aligning Curve/Cylinder	Plastic Sheet	2000	0.003	6.00	Dollarama
	Dowel Aligning Ramp	Al sheet	300	0.009	2.70	Home Hardware
	Dowel Aligning Ramp Supports	1/4" dowel	60	0.0055	0.33	
Reservoir	Frame of Reservoir	Hardboard	784	0.00029	0.23	Home Depot
	Walls	White Corrugated Plastic	2798	0.00147 4496	4.13	Home Hardware
	Wall Separating the two Reservoirs	Acrylic	392	0.00592	2.32	Home Depot
	Ramp in Reservoir	Acrylic	448	0.00592	2.65	Home Depot
Dowel Pusher	Bar for pushing out Dowels	2X1 Framing Lumber	26	0.00422	0.11	Home Depot
	Guides for dowel to fall into place	Al Sheet	160	0.009	1.44	Home Hardware
Conveyor	Actuator	Motor	1	5	5.00	Design Store
	Gears	Plastic Gears	2	0.5	1.00	Active Surplus
	Frame	2X1 Framing Lumber	434.9	0.00422	1.84	Home Depot
	Board for guiding belt	Hardboard	1110	0.00029	0.32	Home Depot

	Belt	Shelf Liner	860.6	0.00055	0.47	Home Hardware
	Dowels for Driving the belt	1” Wooden Dowels	40	0.035	1.40	Home Depot
	Box Assister	Acrylic	405	0.00592	2.40	Home Depot
	Box Assister	Hardboard	286	0.00029	0.08	Home Depot
Box Closer	Railing to Close Boxes	1/4” dowel	21	0.0055	0.12	Home Depot
	Roller to Lock the lid	1” Wooden Dowel	9	0.035	0.32	Home Depot
	Cover of roller to induce rolling	Shelf Liner	71.8	0.00055	0.04	Home Hardware
	Shaft for roller	Al tube	13	0.076	0.99	Home Hardware
	Support of shaft to allow box to fit under the roller	Acrylic	2	0.00592	0.01	Home Depot
Extra Building Material:	L-brackets		23	0.1158	2.66	Home Depot
	#5 Screws		6	0.4	2.40	Home Hardware
	Other Screws/Connectors				10.00	Home Depot and Home Hardware
Electronic Components	Power Supply		1	19.99	19.99	Tigerdirect
	PIC18F4620				5.21	Digikey
	TIP 142		2	1.65	3.30	Digikey
	HCF4050		1	0.56	0.56	Digikey

	Headers				9.00	Various
	Protoboards (various)				7.80	Dipmicro
	Wires				10.00	Creatron
	Capacitors		4	0.04	0.16	Digikey
	74HC14		1	0.56	0.56	Digikey
	Surface Mounts				3.00	Digikey
	1N4001		2	0.48	0.96	Digikey
	SN754410		2	1.66	3.32	Digikey
	Microswitch		3	0.56	1.68	Digikey
	LCD+Keypad				6.00	Design Store
	Resistors (various)		7	0.1	0.70	
Total ($):					**142.04**	

4. DIVISION OF THE PROBLEM

The design of the dowel maker was split into three subsystems: Electromechanical, PIC Microcontroller, and Circuits. Arushri Swarup was responsible for the electromechanical subsystem; Sherry Shi was responsible for the PIC microcontroller subsystem; and Andrew Wong was responsible for the circuit subsystem. Overall, the key design problems were divided as such:

Electromechanical

1. Create the box elevator mechanism to load the boxes onto the conveyor belt
2. Design a gravity-driven funnel system to align and store the dowels
3. Design an effective dowel dispensing mechanism after a dowel has been pushed out by the solenoid.
4. Create the conveyor belt mechanism to move the box from its initial position to the dowel loading position, and finally to the exit area where the box lid will be shut
5. Create a box-closing mechanism using a roller to lock the latch

PIC Microcontroller

1. Interface with the operator
2. Count the number of dowels in each reservoir
3. Detect when the box is in position
4. Load the dowels in the correct position

Circuits

1. Create circuits to transmit signals from the PIC microcontroller to the designated actuators
2. Design circuits to allow the microcontroller to sense when the box is in position as well as count the number of dowels loaded into the reservoir.

5. ELECTROMECHANICAL SUBSYSTEM

5.1 ASSESSMENT OF THE PROBLEM

Design and build mechanisms that perform the following tasks using actuators, motors and solenoids, and gravity as the forces to perform them.

1. Box Loader that would transport empty boxes loaded into the machine to the position where dowels would be loaded into them.
2. Conveyor Mechanism that moves the boxes into position for dowels to be placed into each spot in the box, and when six dowels are loaded, shuts the lid and locks it.
3. Dowel Loader that loads white and dark dowels into the box according to the pattern entered into the machine by the user.
4. Closing Mechanism that shuts the lid of the box and locks the latch.

5.2 SOLUTION

5.2.1 OVERVIEW OF THE PATH OF BOX

The following figures show a visual representation of the path taken by the box when being loaded with dowels by the machine:

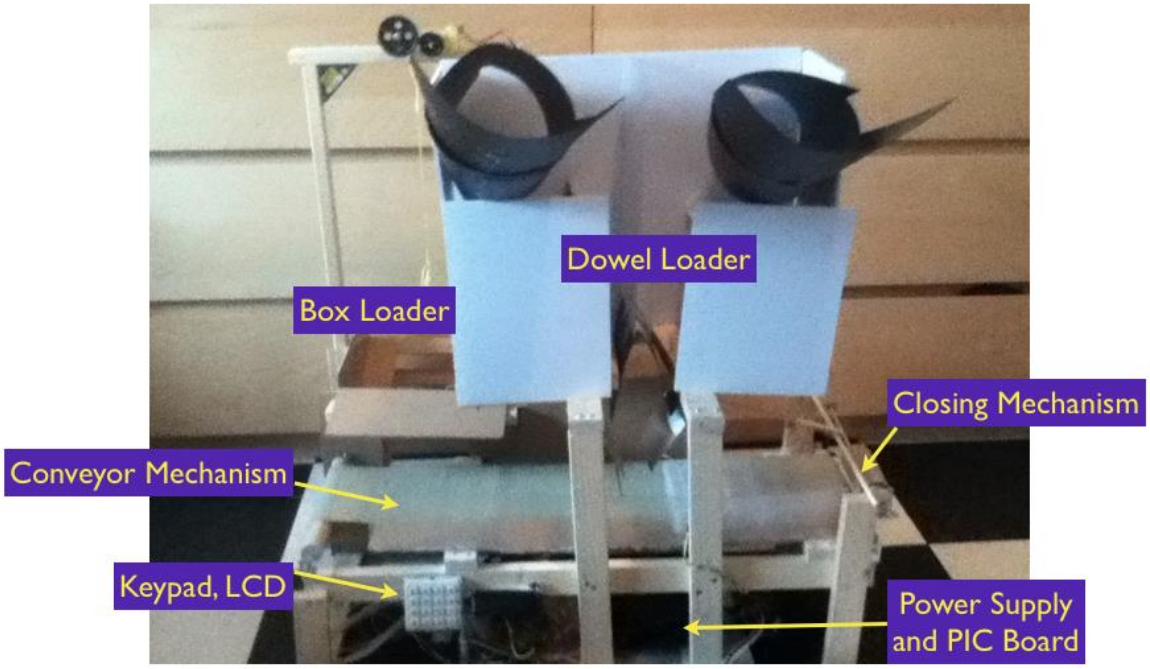


Figure 5-2: Overview of the Machine

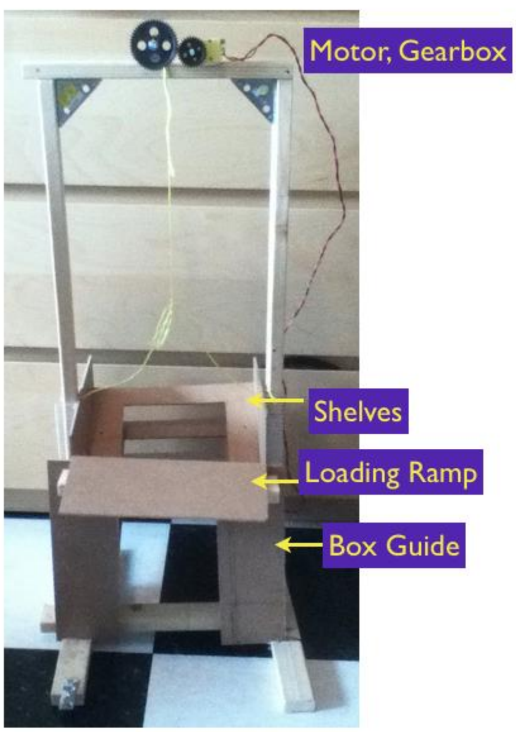


Figure 5-1: Labeled diagram of the frontal view of the box loader. The loading ramp is positioned on top of the conveyor belt.

The user follows the instructions on the LCD and enters in the pattern of dowels to be loaded into the box. The Box Loader loads empty boxes on the Conveyor Mechanism which proceeds to move the box to be loaded with dowels by the Dowel Loader. After the box has been loaded, the Closing Mechanism closes the box lid. Refer to figures 5-1.2 to 5-1.8 in Appendix B for a description of the path the box takes from loading to closing.

5.2.2 BOX LOADER

The empty, open boxes are loaded into the shelves, there is room for three boxes. The box loader uses a pulley system with a motor that turns an axle attached to a string attached to the shelves. The sliding mechanism, made of slippery corrugated plastic, has small friction against the wooden frame. This lessens the amount of friction the motor must overcome to raise the shelves. The angled shelves are made of smooth hardboard, containing the empty boxes. The slippery hardboard surface allows the boxes to slide down easily after the shelf surpasses the box guide while being lifted up by the motor. The shelves on the elevator needed to be as light as possible to allow a 12V DC brush motor, with a gearbox of ratio: 5/3, to lift it. Therefore, minimal material was used to construct it. Much calibration was put into loading the box onto the conveyor after it slides off the elevator. An appropriate angle of the loading ramp on top of the conveyor belt had to be tested for in order to give the box enough momentum to slide to the edge of the conveyor when landing, but not enough for the box lid to shut. This ensured that the box lid was open and in the right position on the conveyor belt for loading dowels. See figure 5-2.1 in Appendix B for a side view.

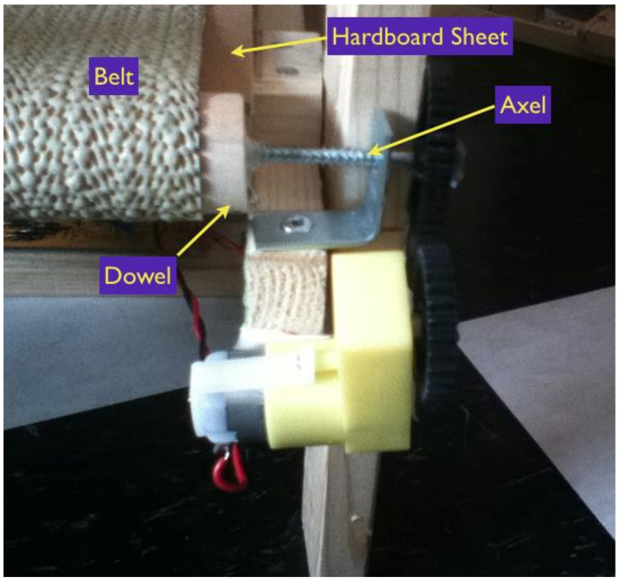


Figure 5-3: Image of the conveyor mechanism.

5.2.3 CONVEYOR MECHANISM

The motor turns the gear attached to the axle which rotates the dowel, to pull the belt along the hardboard sheet. The gear box has a gear ratio of 4/3. The belt material needed to be lightweight, inexpensive and have some friction so the dowel would not simply rotate without moving the belt. The design uses a simple shelf liner which performs the job well. It provides enough friction to be pulled by the rotating dowel but not enough to stop moving when a box is on the conveyor. See figure 5-3 for a detailed image.

5.2.4 DOWEL LOADER

The dowel loader consisted of three subparts: the aligner, reservoir and dowel pusher. Figure 5-4 outlines the dowels' path into the reservoir when loaded. This mechanism was tested to ensure it properly functions when dowels are loaded 12 at a time. Testing of the mechanism in this fashion allowed the team to design the mechanism for dowels traveling while interacting with each other. Dowels traveling together move differently than they do when traveling alone.

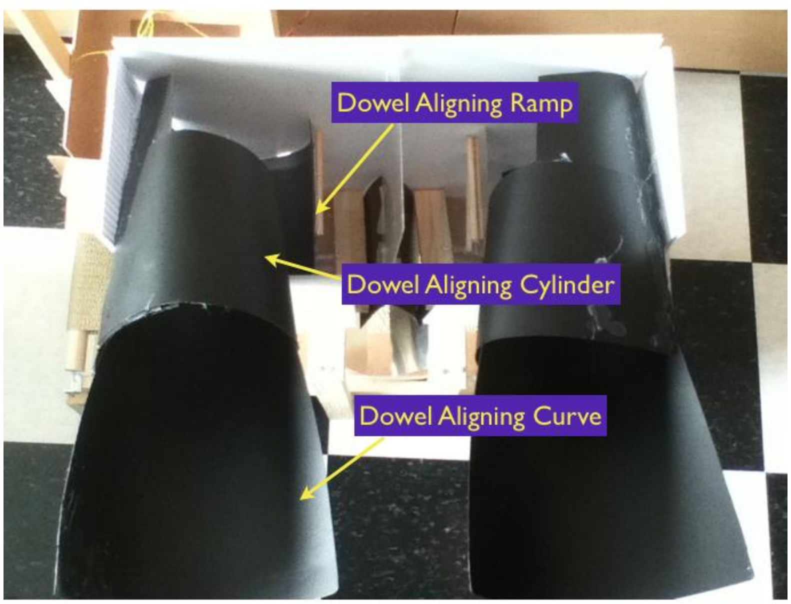


Figure 5-4: The dowels enter the Dowel Aligning Curve; they can be loaded in any way, as per the constraints outlined in the RFP.

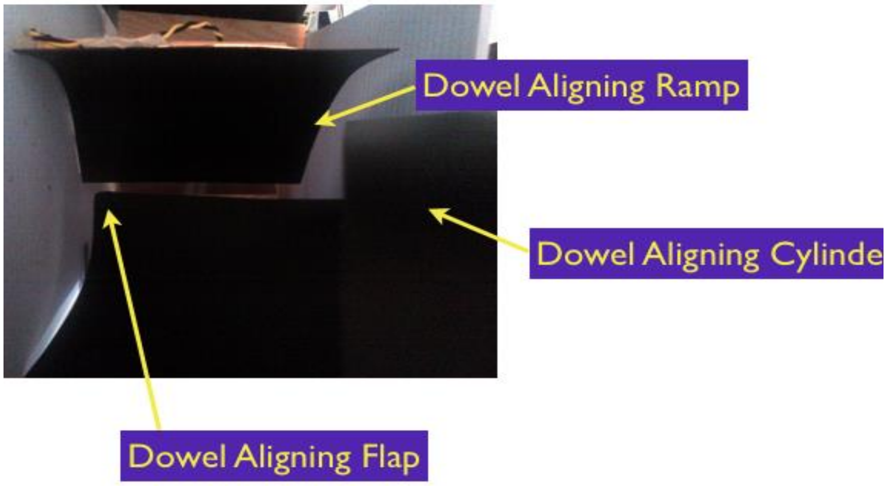


Figure 5-5: The dowels come in contact with the dowel aligning flap which has the same curvature as the cylinder.

The aligner utilizes a lightweight plastic material that interacts with dowels with minimum friction. The plastic was curved into a semi-cylindrical formation, causing dowels to align horizontally when in contact with the surface. When dropped in, the dowels slide down the open surface, spreading apart in the process. They travel down into the dowel aligning cylinder and exit out of the dowel aligning cylinder and then interact with the dowel aligning flap, shown in figure 5-5.

The path is continuous all the way down the dowel aligning mechanism to prevent rotation of dowels, causing misalignment. After the dowel falls down the dowel aligning flap, it contacts the microswitch and is counted by the PIC. Then it rolls down the dowel aligning ramp which then leads to the reservoir ramp on which dowels stack up and roll down into the reservoir. The reservoir, with a capacity of 12 dowels, has an angled bottom, causing dowels to spontaneously roll into position on top of the pushing bar.

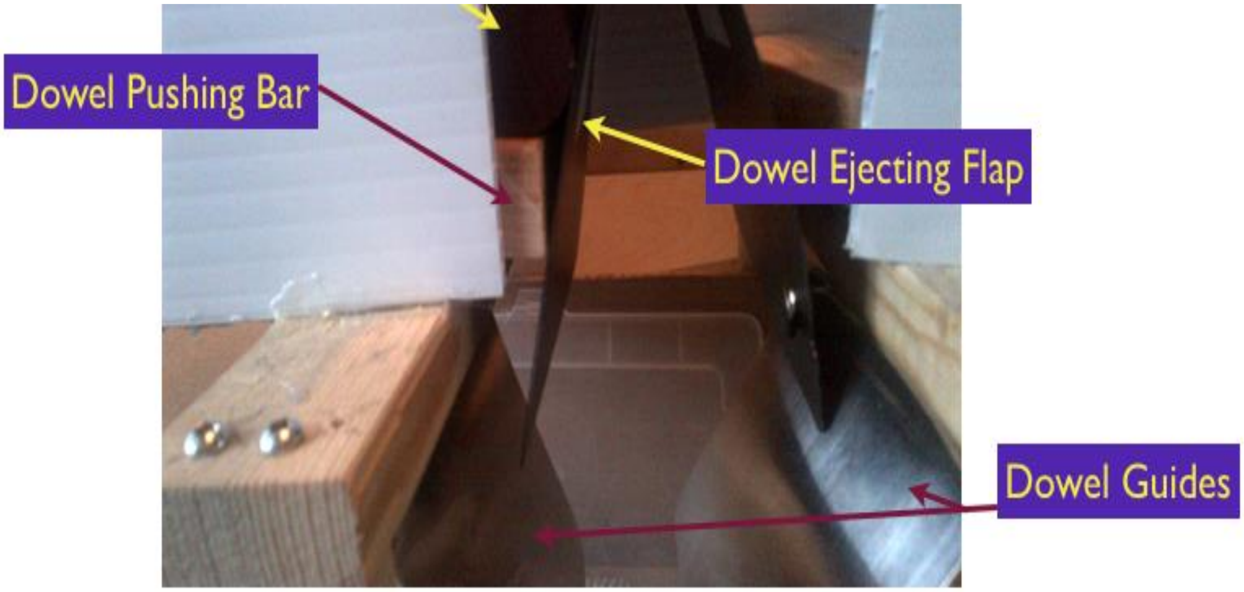


Figure 5-6- This shows the dowels in position, ready to be ejected and loaded into a box.

At this point, the solenoid plunger, attached to the right of the dowel pushing bar, is in release position. When a dowel is ready to be loaded, the solenoid pulls the dowel pushing bar backward, and forward, pushing the dowel out and into the box. The dowel ejecting flap ensures only one dowel is ejected at a time by holding the dowel in place while the dowel pushing bar has been pulled back; the dowel is only ejected once the dowel pushing bar pushes it out, resuming its original release position. The dowel guides ensure that the dowel, no matter which reservoir it is ejected from, loads into one position on the conveyor. See figure 5-6 for more details. To ensure that the plunger did not come out of the socket, a small path was drilled underneath the bar through which a screw would be able to guide the bar. This was effective in restricting the deviation of the bar from its set one dimensional path however did not allow for enough distance for the bar to travel to push out the dowel. A future solution is to instead cut a path out of the side of the reservoir wall and attach a thin dowel to each end of the bar to travel along that path to allow for a variable distance the bar can travel.

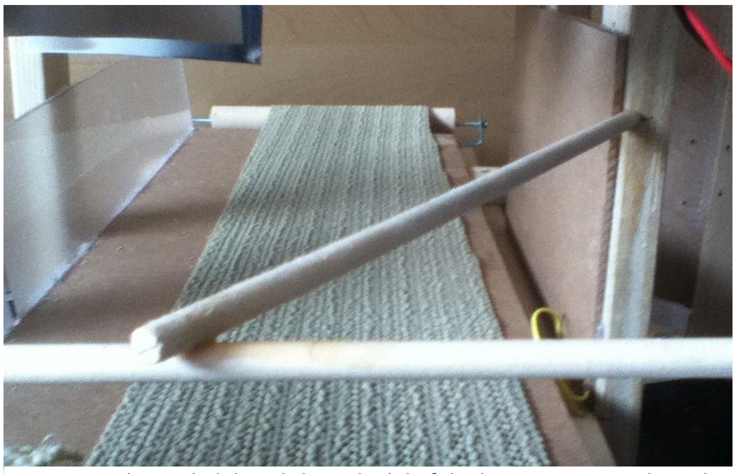


Figure 5-7: The angled dowel shuts the lid of the box as it moves along the conveyor belt

5.2.5 Closing Mechanism

After the lid is closed using the mechanism in figure 5-7, the box passes underneath the roller, which is covered by the same material as the conveyor belt. The material creates enough friction between the box and the roller so the box does not simply slide underneath the roller; it rolls across the box to shut the latch. The hopper allows the axle of the roller to vertically move up and down, allowing the box to fit underneath the roller, without putting a stop to the movement of the conveyor as shown in figure 5-8.

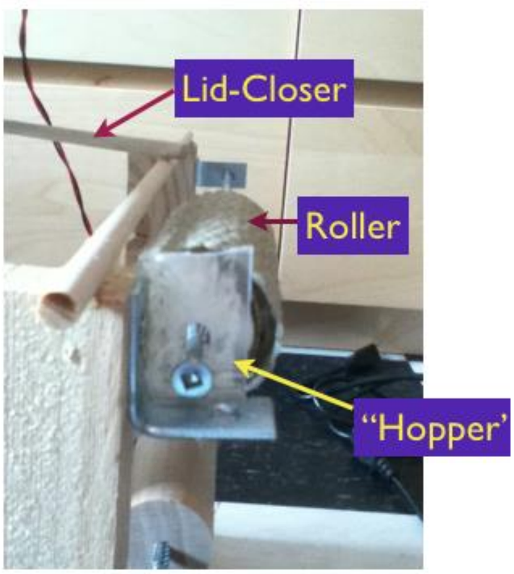


Figure 5-8: This mechanism locks the latch of the box.

5.2.6 Actuators, Sensors, Keypad, and LCD

See figures 5-9.1 to 5-9.7 in Appendix B for details about the placements of the actuators, sensors, keypad and LCD. See figures 5-9.1 and 5-9.2 for details about the rotary actuators. Figures 5.9-3 and 5-9.4 show the placement of the solenoids on the machine, there are two solenoids, one for each dowel reservoir. Figures 5-9.5 and 5-9.6 show the location of the three microswitches. Lastly figure 5.9-7 shows the location of the keypad and LCD. 20 g microswitches were used, as they were sensitive enough to detect a slight movement; practical for detecting the top of an empty box and detecting a lightweight 25g dowel.

5.2.7 Box Assisters

Please refer to figures 5-10.1 to 5-10.3 in Appendix B which explains the various mechanisms which keep the box in place during operation.

5.2.8 DESIGN FOR: WEIGHT, COMPACTNESS, AND COST

Weight: The total mass of the machine was not to exceed 6 kg. This put a constraint on the materials used and therefore a constraint on the reliability of the electromechanical components of the machine. Lightweight material, namely hardboard and 2 by 1 framing lumber, was largely used to construct a design meeting the weight requirement. Had the weight limit been increased, metal, a more reliable, and structurally sound material, could have been utilized.

Compactness: The machine was not to exceed spatial dimensions of a cube 75 cm in length. This put a constraint on limited surface area and problems were encountered when designing the dowel loader. A greater surface area on the aligner mechanism would have allowed the dowels to spread out before entering the reservoir, causing it to be a more reliable mechanism.

Cost: The total cost of the machine was not to exceed $250.00. The budget for the circuitry was reallocated to electromechanical material. This lead to the design decision of constructing a PIC board by hand instead of using the DevBugger, which was $50.00. This saved money however it was difficult to make the circuitry completely reliable because of the design of the LCD which required short wires because of signal degradation.

5.3 SUPPORTING CALCULATIONS

5.3.1 TORQUE CALCULATIONS

Mechanism	m (kg), g (m/s2), r (m)	τ (Nm)
Box-Loader	0.963, 9.81, 0.0025	0.024
Conveyor	0.430, 9.81, 0.0127	0.054

The mass used to calculate torque needed by the Box-Loader motor includes that of the shelves, 610 g, three empty boxes, 345g, and the sliding guide, 8g. The mass used to calculate torque needed by the Conveyor motor includes that of two boxes loaded with dowels, 380g, and the belt, 50 g. The DC Brush motor chosen to drive the two mechanisms was Shenzhen DC Brush Gear Motor TGP01S-A130 with an output torque of 0.066 Nm which is more than enough torque to drive the mechanisms.

A gear box was used to amplify the torque supplied by the motors to ensure that enough torque was supplied to the mechanism, in case of the presence of uncalculated forces needed to be overcome by the motor, ie. friction. This made the mechanism more reliable.

5.3.2 GEAR RATIO CALCULATIONS

Mechanism	d_l, d_m (cm)	η	τ_{output}(Nm)
Box-Loader	5, 3	5/3	0.11
Conveyor	4, 3	4/3	0.088

τ_{output} was calculated using $\eta = \frac{\tau_{\text{output}}}{\tau_{input}}$. Therefore, the torque supplied by the gear box of the box-loader was 0.11 Nm and the torque supplied by the conveyor gear box was 0.088 Nm.

5.3.3 LINEAR FORCE CALCULATIONS

Force required by the solenoid to push out a dowel:

$$\begin{aligned} F &= ma \\ &= (m_{dowel} + m_{dowel\ pushing\ bar})g \\ &= (0.025\ \text{kg} + 60.19\ \text{kg}) \times 9.81\frac{m}{s^2} \\ &= 0.836\ \text{N} \end{aligned}$$

The solenoid used on the machine is a CII/A1464 solenoid item number G16036, refer to Appendix E. This solenoid has a pull force of 3.7 lbs, or 1.7 kg which is more than enough force to eject a dowel. This solenoid was chosen because of its accessibility from the purchased design kit and its inexpensive price.

5.4 SUGGESTIONS FOR IMPROVEMENT OF THE SUBSYSTEM

5.4.1 THE DOWEL LOADER: DOWEL ALIGNER VIBRATOR

The dowel aligner mechanism worked about 80% of the time while testing it. When loading dowels in too fast, or too many at a time, one or two dowels get stuck. If the aligners were moved around a little, the dowels would move around and become unstuck. In order to make it a more reliable dowel aligner, the design would include a motor attached in between the two aligners. The motor, connected to a large enough gear to touch each aligner, would rotate the gear and cause the aligners to vibrate. The vibrations would urge the dowels to move and become unstuck. The installation of the "vibrator" would have to be calibrated so that the aligners would move unidirectionally from side to side. Otherwise, dowels would misalign.

Figure 5-9: This shows the bottom of the dowel pushing bar, and the screw protruding out from underneath it. This screw follows a channel drilled in the bottom of the reservoir.

5.4.2 THE DOWEL LOADER: DOWEL PUSHING GUIDE

Currently, the dowel pushing bar has a screw attached to it which follows a channel drilled into the bottom of the reservoir, refer to figure 5-9. This channel ensures the plunger does not come out of the solenoid socket. The length of the channel is restricted by the body of the solenoid; when the solenoid tries to push out a dowel, the screw-channel path restricts the distance traveled by the bar, preventing a dowel from being ejected. Therefore, instead of having a screw follow the channel on the bottom of the reservoir, a screw or thin dowel would be attached to the side of the pushing bar and a path would be carved into the wall of the reservoir. This would allow the bar to be pushed a greater distance.

6. CIRCUITS SUBSYSTEM

6.1 ASSESSMENT OF THE PROBLEM

The purpose of this system is to allow communication between the PIC microcontroller and the electromechanical actuators and sensors in the machine. A custom built circuit board was used to connect the user-interface peripherals, the LCD display and keypad, and the actuators and sensors, DC motors, solenoids and microswitches, to the PIC microcontroller. Furthermore, the circuits subsystem required an emergency stop switch which would stop every actuator when needed to ensure the safety of its operators.

6.2 SOLUTION

Note that circuit diagrams for the motor control, solenoid control and microswitch circuits can be found in Appendix B, figures 6-4 to 6-6.

6.2.1 PIC MICROCONTROLLER CIRCUIT

The PIC Microcontroller circuit acted as a hub for the interface between the microcontroller and its peripherals. A power distribution section was also present which acted as an interface between the power supply's 24-pin connection and the various elements in the circuit subsystem. Due to the sensitive nature of the microcontroller, it was given a clean voltage source directly from the power supply to reduce the chance of electrical interference. Furthermore, to prevent negative feedback from the rest of the circuit, all of the microcontroller output pins were connected to a ⅙ HCF4050 buffer. MCLR on the microcontroller was connected to the PWR_OK signal from the power supply which indicated when the power supply was in a stable state. This ensured that the entire circuit subsystem was connected to a reliable source of power before starting any operation. The keypad was interfaced with a MM74C922N decoder with 0.01 and 0.1 µF capacitors which act as the clock. The LCD was initially interfaced with the microcontroller via a HCF4050 buffer. However, due to issues with signal degradation, the

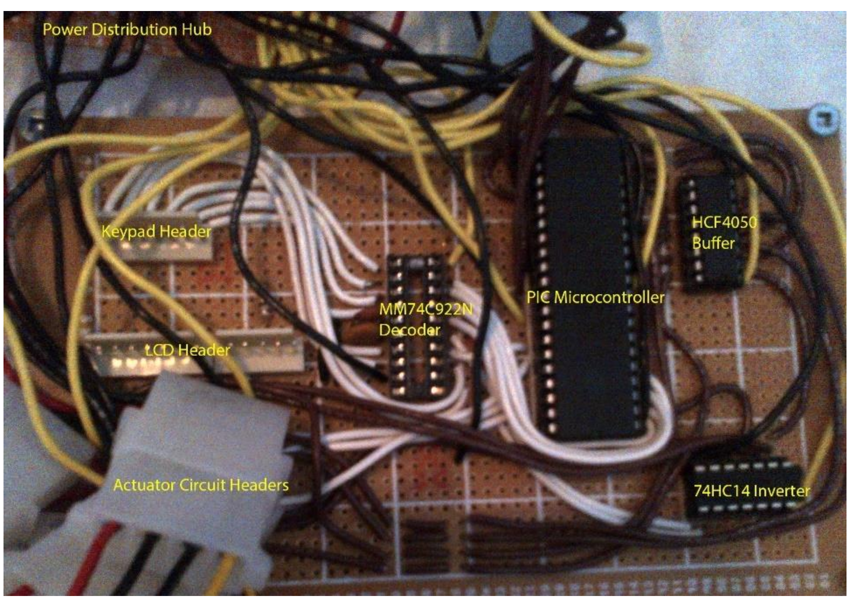


Figure 6-1: PIC Microcontroller Circuit

buffer was removed. All of the peripherals were connected to this circuit via headers to allow for easy removal or troubleshooting. See figure 6-1 and 6-4 for more details.

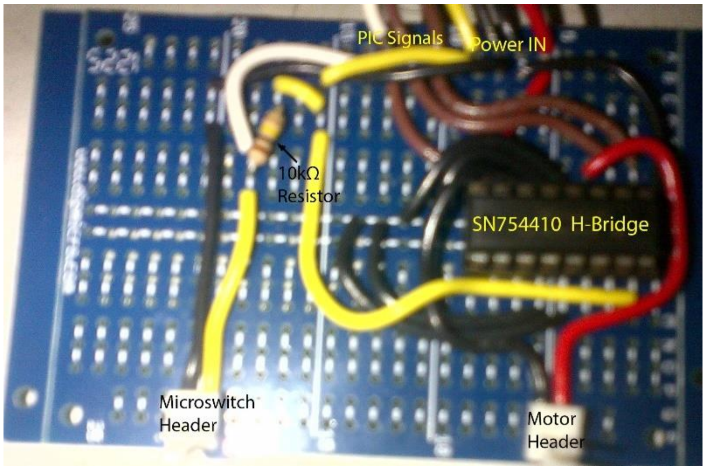


Figure 6-2: DC Motor driver and associated microswitch circuit.

6.2.2 DC Motor Driver

Two DC motors were used in the final design: one to power the box loader, and the other to power the conveyor. The motors used were Shenzhen DC gear motors, which provided low speed and high torque which was optimal for their designated tasks. The motors were individually controlled using a pair of Texas Instruments SN754410 H-bridge drivers. This allowed the PIC microcontroller to change rotation speed via PWM, and the direction of the motor with ease. In addition, to ensure that the motor had enough power to reach its maximum speed or torque, it was run at 12V, or its maximum rated voltage. PW M was used to slow the motor down in the event it ran faster than necessary. Two outputs from the microcontroller were needed, an enable signal to turn the motors on and off via PWM, and a direction pin. Since the SN754410 needed two opposing direction signals, a ⅙ 74HC04 HEX inverter was used. The schematic of the circuit is shown in Appendix B, figure 6-5. To prevent AC current feedback from the motor to the driver, a 1µF capacitor was soldered between the motor leads.

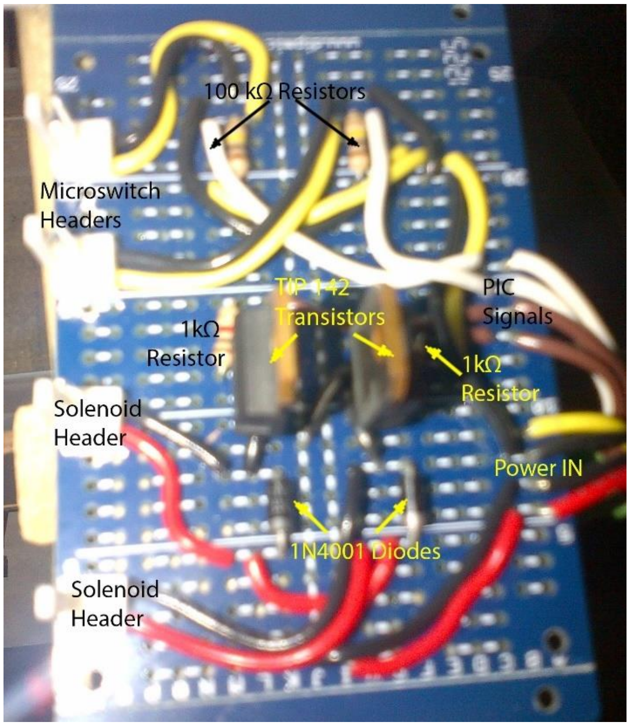


Figure 6-3: Solenoid Driver with associated microswitch circuits for dowel counting.

6.2.3 Solenoid Driver

Two solenoids were used in the final design to push the dowels from the reservoir into the box. The solenoids were activated using a TIP-142 Darlington transistor and a 1N4001 protection diode to prevent any backward current from the solenoid. The TIP142 transistor was chosen as it allows up to 10A of current to safely pass, ensuring that the transistor will not burn out or overheat during operation. A 1kΩ resistor was used at the base, giving a base current I_B of 5mA or a collector current of up to 5A which is sufficient to run the solenoid which requires at most 3A.

Supporting calculation:

$$I_C = H_{FE} \times I_B$$

$$
\begin{aligned}
&= H_{FE} \times \frac{V_B}{R_B} \\
&= 1000 \times \frac{5\ \mathrm{V}}{1\ \mathrm{k\Omega}} \\
&= 5\ \mathrm{A} > 3\ \mathrm{A}
\end{aligned}
$$

The schematic of the circuit is shown in Appendix B, figure 6-6.

6.2.4 Sensor Circuits

To keep the design simple, 20g microswitches were used as sensors to detect when the box was in position and to count the number of dowels in the reservoir as they entered. A 100 kΩ pull-up resistor was used to interface the switches with the microcontroller to ensure that the input current does not exceed the maximum rated current of 25mA.

Supporting calculation:

$$
\begin{aligned}
I &= \frac{V}{R} \\
&= \frac{5\ \mathrm{V}}{100\ \mathrm{k\Omega}} \\
&= 0.01\ \mathrm{mA} < 25\ \mathrm{mA}
\end{aligned}
$$

When the switch was in the off position, the microcontroller will read a high signal. Alternatively, when the switch is activated, the microcontroller will read a low signal. The schematic of the circuit is shown in figure 6-7.

6.2.5 Power Supply

To ensure that all actuators received enough power, a computer ATX power supply was used, specifically the Coolmax I-500 [5]. This 500W power supply allowed up to 34A at 12V, which was more than enough to run two motors and two solenoids in parallel at the same time. Furthermore, this type of power supply contains several terminals of the same voltages. This allowed for the power isolation of circuits, thereby reducing the possibility of interferences from other circuits or actuators. In addition, this power supply contains a PWR_OK signal, which was interfaced with the PIC microcontroller. When the power supply detected a stable power source, it would send a high signal to the microcontroller, instructing it to begin operations.

To enable the power supply, the PWR_ON signal must be grounded. Table 6-# shows the specifications of the power supply.

Table 6-1: Coolmax I-500 Power Supply maximum current specifications. *[5]*

	+5V	+12V1	Ground	PWR_ON	PWR_OK
Max. Output Current	21A	34A	N/A	N/A	N/A
Colour	Red	Yellow	Black	Green	Purple

6.2.6 EMERGENCY STOP SWITCH

An emergency stop switch was implemented to cut off all power to the actuators, but not to the PIC microcontroller. The initial design was to use a ULN2001A transistor array as a means to stop all output signals from the microcontroller to their respective destinations. However, after further research and testing, it was found to be an unfeasible solution. As such, a simpler design was chosen. A switch was connected from the 12V power to the actuator circuits. When the switch is in the on position, current would flow to the actuators, which allows the actuators to function. When the switch is in the off position, no current would flow to the actuators, resulting in the actuators stopping. This way, the microcontroller would still function, but actuator would stop, ensuring the safety of the machine operator.

6.3 SUGGESTIONS FOR IMPROVEMENT OF THE SUBSYSTEM

The circuit subsystem could have been improved in the following ways:

1. Consolidating all the actuator drivers into a single circuit

Currently the actuators and sensors span 3 circuit boards, bundled only together by their proximity. By placing consolidating the motor driver circuits, a single SN754410 H-bridge driver could have been used, lowering cost. Furthermore, it would have been easier to troubleshoot the driver circuits. One drawback from this would be that the extended length of the wire may act as an antenna for EM interference.

2. More secure connectors

The current method used a mix of 18AWG Molex connectors and 22AWG low-quality pin headers. By using a uniform system of higher quality pin headers, it would prevent loose or poorly made connections.

7. PIC MICROCONTROLLER SUBSYSTEM

The automation of the system is made possible by a microcontroller, which is responsible for sending, receiving and processing signals to and from the electrical components. The microcontroller used in this system is the PIC18F4620, which runs assembly language code to communicate with two DC motors, two solenoids, and three microswitches. In addition, the microcontroller must communicate with the user through an LCD-keypad user interface. A full pin-by-pin description is available in table 7-1 in Appendix B.

7.1 ASSESSMENT OF THE PROBLEM

7.1.1 1INTERACTING WITH THE USER

The user interface should efficiently and effectively greet the user and gather the necessary information for the operation. The user must be able to provide the number of boxes to be packed and the pattern corresponding to each box. There should also be a “Start” button which will initiate the operation when the machine is ready. After the operation, the user interface must report the amount of time elapsed for the operation and display the packed patterns.

7.1.2 DATA PROCESSING AND LOGGING

Upon receiving the packing information from the user, the microcontroller must be able to process the information during the operation and pack the boxes according to the user's specifications. The microcontroller should also log the packaging information recorded during each operation, including number of boxes packed, the pattern corresponding to each box, and the operation time.

7.1.3 RECEIVING INPUT SIGNAL

The microcontroller should effectively receive input signals from the three microswitches - two of which counts the number of dowels that are in the reservoir and one that detects when the box is in position on the conveyor belt.

7.1.4 SENDING OUTPUT SIGNAL / MOTOR CONTROL

During the operation, the microcontroller must control the two DC motors and the two solenoids.

7.1.5 EMERGENCY PROTOCOLS

An emergency stop switch must be included in the system. Upon receiving a signal from the emergency stop, the microcontroller must stop all actuators until the emergency stop flag is cleared, upon which time resume the operation by setting the actuator to their previous states.

7.2 SOLUTION

7.2.1 USER INTERFACE

The user interface is composed of a HD44780U dot-matrix 16x2 LCD and a 4x4 keypad. The main program initiates to an "Idle" stage, in which a greeting message would prompt the user to initialize the operation. The user is then given the choice of either viewing the operation logs or to start a new operation. Upon selecting the "Pack Dowels" option, the user is then prompted to select the number of boxes (up to 3) and the pattern (1 - 6) corresponding to each box. After confirming the choices, the user can then press 'A' to start the operation.

The DATA_OUT pins of the keypad are connected to PORTA (see Appendix B, Table 7-1 for detailed pinout table), while the DATA_AVAILABLE pin is connected to PORTB. This allows the keypad to either interrupt or be polled to read the input data. In this case, the keypad is continuously polled through a loop when waiting for user input.

The LCD is connected to PORTD and operates in the 4-data-pin mode (only D4:7 are used for writing). The R/W pin is connected directly to ground since only WRITE instructions are executed.

7.2.2 DATA PROCESSING

After each operation, the packing information, including operation time, the number of boxes packed and the pattern corresponding to each box, are stored in the EEPROM. 8 bytes are used to for each run (4 bytes for the 4 digits of the operation time, 1 byte for the box count, and 3 bytes for up to 3 patterns). The log starts at address 0x00, and the variable, EE_Adr_current, points to the next empty EEPROM address.

7.2.3 RECEIVING INPUT SIGNALS

The microswitches used to count the number of dowels in each reservoir are connected to RB4 and RB5. This allows them to interrupt the main program upon a change of state. To ensure the counts are not incremented on both the rising and falling edges, he ISR checks if the bits are clear, upon which the variables dowel_count_L and dowel_count_R will be incremented. The conveyer belt microswitch is connected to the External Interrupt 2 pin (RB2), which interrupts on the falling edge to signal the arrival of the box.

7.2.4 OUTPUT SIGNALS

The outputs of the two DC motors are connected to PORTC. The PIC microcontroller controls the rotation speed through PWM and the direction through a high or low signal. The PWM is implemented using the CCPR1 and CCPR2 modules. The supporting calculations are shown below:

PWM Calculations

Oscillator Period (T_{osc}) = 1/8MHz = 0.000000125 seconds
PR2 = b'01100011' = 99
TMR2 Prescale Value = 4

$$PWM\ Period = 4 \times T_{osc} \times (PR2 + 1) \times (TMR2\ Prescale\ Value) \quad [Seconds]$$

$$= 4 \times 0.000000125 \times 100 \times 4 = 0.0002\ seconds$$

$$PWM\ Frequency = \frac{1}{PWM\ Period} \quad [Hz]$$

$$= \frac{1}{0.0002\ seconds} = 5.0kHz$$

7.2.5 EMERGENCY PROTOCOLS

The emergency stop button is connected to RB0, which allows for high-priority interrupts only. When the emergency stop is pressed, the emergency stop flag, EStopF, is toggled. The ISR checks if the flag is raised or dropped. If the flag bit is high (raised), the ISR stops all actuators and store their current states. The emergency stop interrupt is re-enabled in the ISR, and the program executes an infinite loop. When the emergency stop is pressed again, the flag bit will be cleared and all actuator states will be restored.

7.2.6 ADDITIONAL SOLUTIONS

Other than the solutions described above, the following components are necessary for the operation of the program.

7.2.6.1 TIMER0

The operation is timed using the TIMER0 module, which is enabled when the user presses “Start” and stopped at the end of the packaging operation. The calculations for the TIMER0 configurations are shown below:

Timer calculations

Oscillator Frequency: 8MHz
Prescaler: 1:64
Preload value: 6
Mode: 8-bit
Timer0 overflows from FFh (255) to 00h

$$Timer\ Tick\ Frequency = \frac{(Oscillator\ Frequency)}{4} = \frac{8MHz}{4} = 2MHz$$

$$Interrupt\ Frequency = \frac{Tick\ Frequency}{Prescaler \times (Overflow\ Value + 1 - Preload\ Value)}$$

$$Interrupt\ Frequency = \frac{2 \times 10^6 Hz}{64 \times (255 + 1 - 6)} = 125Hz$$

7.2.6.2 DEBOUNCING

The three microswitches and the emergency stop are debounced through software. The debouncing algorithm, which is implemented using a macro, polls the input signal (specified by Button_reg and Button_bit) every 5 ms. If the signal is high, the “Pressed Confidence Level” (Pressed_Clevel) is increased, and the “Released Confidence Level” (Released_Clevel) is cleared, and vice versa if the signal is low. When either of the confidence levels exceeds a threshold level, 10 in this case, the button is considered pressed or released, and the variable Pressed will be set to either 1 or 0, respectively. This means that the button must be pressed for at least 50 ms for the signal to pass through.

7.3 SUGGESTIONS FOR IMPROVEMENT OF THE SUBSYSTEM

The microcontroller subsystem can be improved in the following ways:

1. Organization - Currently, the entire program is written in one file. With over 1000 lines of code, this makes the program difficult to navigate through, especially during debugging. Separating the large file into multiple small files would allow for more organization and ease of navigation. The organization can be further improved with constancy in variable names and more commenting.

2. Set a maximum number of logs - After each operation, the data is stored at the EEPROM address specified by the variable, EE_Adr_current, which is an 8-bit value that increments upon each "write" operation. There is no method of limiting the amount logs stored in the EEPROM, which means the memory will be overwritten after the 256th "write". However, this would only be problematic after 32 operations without turning off the machine (since the program is reset at each power-on), which is rare.
3. Store operation time in one byte to save EEPROM usage - Currently, the operation time is stored in four bytes (one for each digit). This can be reduced to two bytes, if each digit is stored using the four most significant and the four least significant bits of each byte.
4. Real Time Clock (RTC) - The RTC was not implemented, as it would require an external driver circuit, which would increase the complexity risks for bugs in the PIC hub. However, a RTC would be very beneficial to the microcontroller subsystem, as it would allow for real time and date display, and can act as an external clock source, which may be more reliable than the internal clock source.

8. Integration and System Improvement Suggestions

Due to several design changes which resulted in delays, full system integration and testing was not fully completed in time. During partial integration, numerous issues were found, which are outlined in the following subsections.

8.1 Non-Functional LCD

Debugging Process: To find the location of the problem, a multimeter was used to check for shorted connections. The resistance between two areas of the circuit were measured; if resistance was zero, the connection was shorted.

Problem 1: included loose connections.

Solution: Re-soldering the loose connections caused the backlight to turn on and characters to be displayed.

Problem 2: The characters displayed were unrecognizable, and were a result of a degrading signal from the PIC to the LCD.

Solution 1: The wires connecting the PIC to the LCD would have to be shortened. When tested, this did make characters move on the screen. Therefore long wires were in fact degrading the signal, causing junk characters to show up. This could have been fixed by hardwiring the LCD to the PIC and allowing for the shortest possible wires. However, this still did not give recognizable signals, therefore a future solution is to utilize the DevBugger as the PIC board instead.
Source: Professor Emami

Solution 2: Lowering the clock frequency of the PIC from 32 to 8 MHz. Since the signals would be travelling slower, there would a lower chance of signal reflections causing distortions or degradation of the source signal. However when testing, this did not alter the functionality of the LCD.

Problem 3: The LCD would heat up very high almost as soon as the power supply was turned on.

Solution: The power supply unit might have been the cause of the problem as it did not supply the correct voltages. The PIC power, which was supposed to be 5 V showed around 10 V. As such, the LCD would be receiving voltages well above its operating specification, potentially causing the LCD to burn out. This cause could only be tested by implementing a new power supply and checking if the LCD worked then. However, by the time this was realized, the LCD module was burnt out, rendering that specific part useless.

Next Steps: The non-functionality of the LCD could have been avoided by not making the design decision to fabricate a PIC board by just using wires and solder due to the high chance of error. A more reliable solution would have been to use the DevBugger or to fabricate a printed circuit board instead.

8.2 Motors not Functioning

Problem 1: H-bridge burned out on the circuit. This was determined by testing the faulty H-bridge on the functional circuit, which made said circuit non-functional.

Solution: Replaced the faulty H-bridge with a new one.

Problem 2: Wires connecting the circuit to the power supply and control from the PIC were "frayed": there were multiple threads of metal running through the same insulation, making one connection. The small threads came into contact, resulting in a short circuit. This cause of problem was confirmed by the visualization of sparks seen between these connections when the power supply fed power into the circuit.

Solution: Heat shrink wrap was used to connect a single metal thread wire to the frayed ones and soldered to the circuit to minimize places of contact for a short circuit.

Problem 3: Motor functional for approximately 2-3 seconds while connected to power/PIC. This was probably caused by the power supply not supplying the correct voltage. Note that this could also have been the cause of problem 1. This was tested by powering the motor with the provided power supply.

Solution: A more reliable power supply would need to be used.

Next Steps: Based on what is currently known, a more reliable power supply is needed to prevent damage to other circuit components. Similar to section 8.1, a fabricated printed circuit board would have been a more reliable solution to prevent shorted connections.

8.3 BOX LOADER UNABLE TO BE LIFTED

Problem 1: Drawer mechanism was too sticky. The motor could not overcome the friction between the mechanisms to lift the boxes

Solution: The drawer mechanism was removed and replaced with corrugated plastic to guide the boxes up the elevator.

Problem 2: Box container was too heavy. Even with the reduced friction, the motor could not generate enough torque to lift the boxes.

Solution: The entire system was rebuilt to make the entire mechanism lighter. Firstly, glue was used as an adhesive instead of metal L-brackets. Secondly, over 50% of each shelf was removed, leaving only a U shape area for the box to rest in. Lastly, a larger gear was used to increase the torque of the motor.

8.4 REDESIGN OF THE DOWEL LOADING MECHANISM

Problem 1: A single cylinder in the middle of the two reservoirs, as in the proposed design, was not reliable to align dowels; caused dowels to get stuck.

Solution: A design utilizing a curved surface encouraged alignment of dowels. After testing with a few materials, a smooth plastic sheet arranged to form a curved surface seemed to align dowels nicely, no matter what formation they were dropped in.

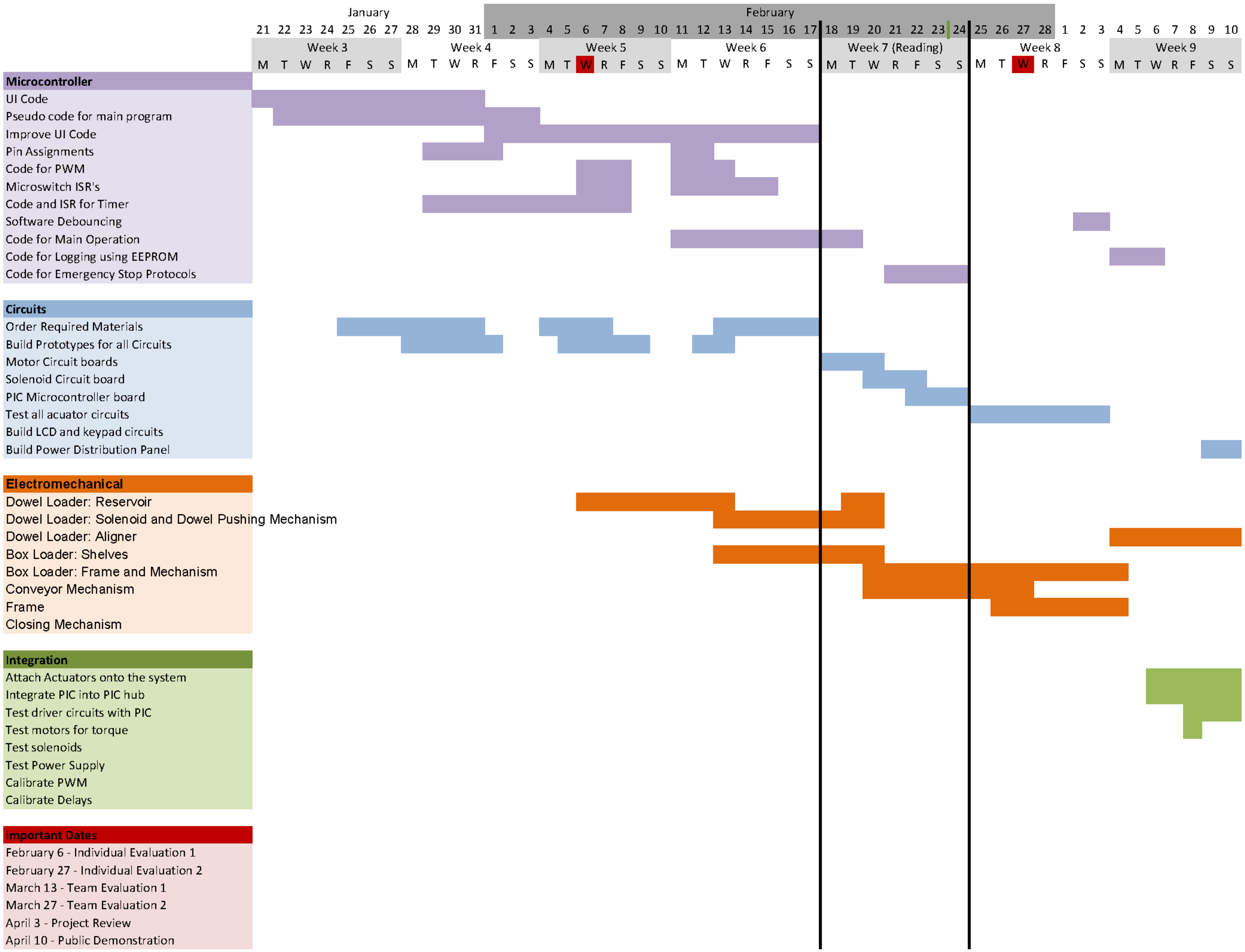
Problem 2: A small surface area in the channels caused dowels to get stuck when many (roughly more than 8) were put in at once.

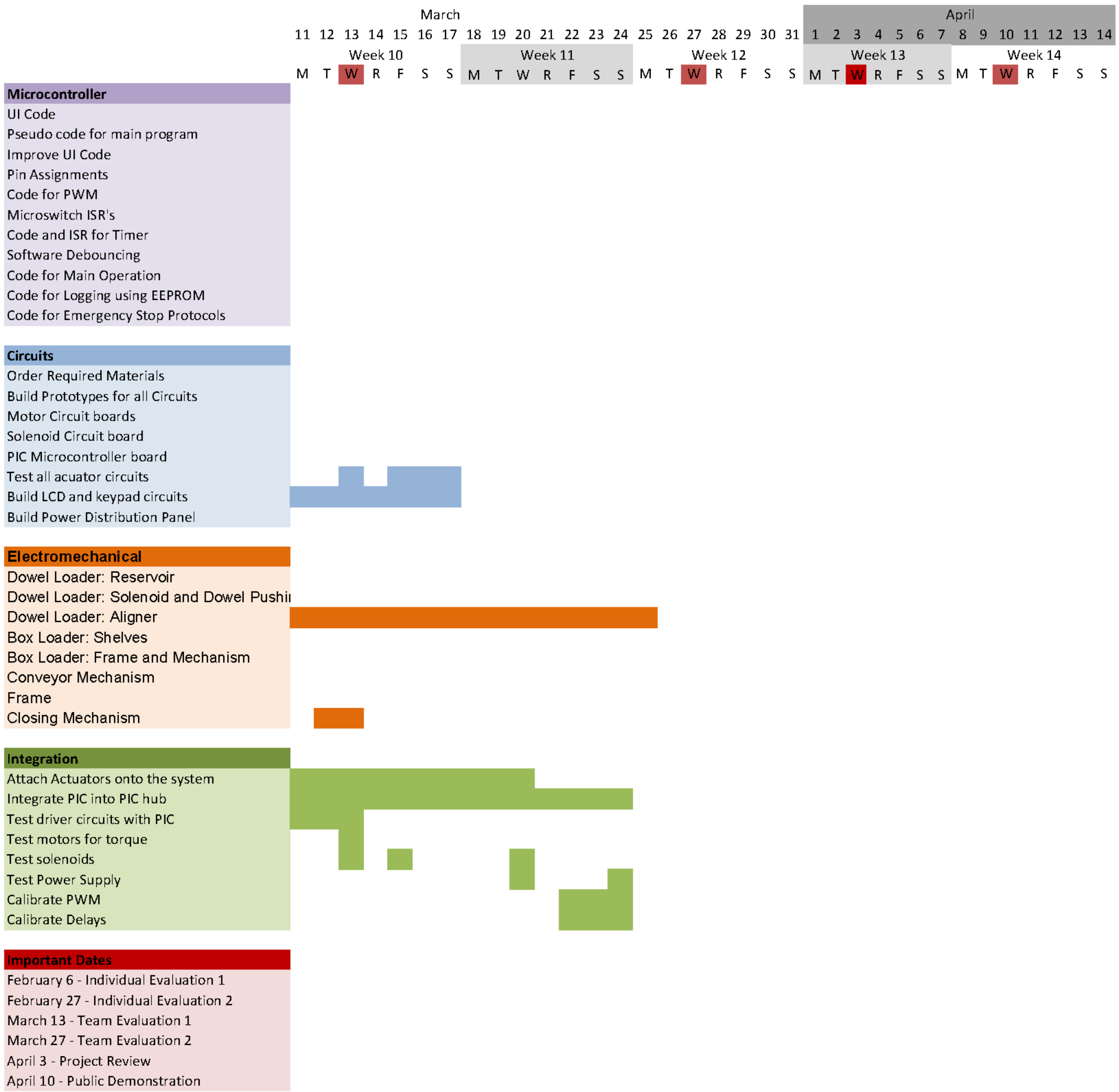
Solution 1: Raising the angle allowed the dowels to overcome static friction more easily.

Solution 2: Extending the length of the channels allowed the dowels to separate as they fall.

Next Steps: As suggested in section 5.4.1, adding a vibrator motor would assist in ensuring that the dowels do not get stuck.

9. ACCOMPLISHED SCHEDULE




March
April
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Week 10
Week 11
Week 12
Week 13
Week 14
M T W R F S S M T W R F S S M T W R F S S M T W R F S S M T W R F S S
Microcontroller
UI Code
Pseudo code for main program
Improve UI Code
Pin Assignments
Code for PWM
Microswitch ISR's
Code and ISR for Timer
Software Debouncing
Code for Main Operation
Code for Logging using EEPROM
Code for Emergency Stop Protocols
Circuits
Order Required Materials
Build Prototypes for all Circuits
Motor Circuit boards
Solenoid Circuit board
PIC Microcontroller board
Test all acuator circuits
Build LCD and keypad circuits
Build Power Distribution Panel
Electromechanical
Dowel Loader: Reservoir
Dowel Loader: Solenoid and Dowel Pushi
Dowel Loader: Aligner
Box Loader: Shelves
Box Loader: Frame and Mechanism
Conveyor Mechanism
Frame
Closing Mechanism
Integration
Attach Actuators onto the system
Integrate PIC into PIC hub
Test driver circuits with PIC
Test motors for torque
Test solenoids
Test Power Supply
Calibrate PWM
Calibrate Delays
Important Dates
February 6 - Individual Evaluation 1
February 27 - Individual Evaluation 2
March 13 - Team Evaluation 1
March 27 - Team Evaluation 2
April 3 - Project Review
April 10 - Public Demonstration

10. CONCLUSION

This design project, “Wood Packer”, attempts to respond to the RFP which requested for a dowel packaging system that can pack two boxes of six dowels according to user specification in less than 2 minutes. Despite best efforts, the project was not successful, mainly due to unforeseen problems encountered during the integration of the circuits and the microcontroller. The modular approach to this design allowed us to pinpoint the root of the problem to be a malfunctioning power supply.

Despite the overall failure, there were many successes in the design. For example, the choice to use wood for most of the mechanical components allowed for a lightweight yet robust structure. The system consists of only four actuators - two DC motors and two solenoids - which simplifies the construction of the microcontroller and circuit subsystems. Modularity was a major design strategy adopted by all subsystems, which allows for easy testing and debugging. Furthermore, the design is very cost-efficient, valuing at a total of $142.02, almost $90 under the cost limit of $230.

Assuming that the current design is successful, many improvements can still be made. A major limitation to the reliability in the design is the reservoir. The current design relies on gravity to guide the dowels in the desired orientation. However, this becomes extremely unreliable when many dowels are loaded at once, especially the dark dowels, which have a rougher surface and tend to get stuck. Integrating an actuator in the design would have greatly improved the reliability. Another major area of improvement is to include a RTC, which would allow for real time and date display and an external, more reliable clock source. Finally, loose connections caused a lot of problems during integration, thus higher quality connectors in the circuits is a worthy investment.

11. REFERENCES AND BIBLIOGRAPHY

11.1 REFERENCES

[1] S. R. Rao, "Advantages and Disadvantages of Automation," 17 Febuary 2006. [Online]. Available: http://www.citeman.com/205-advantages-and-disadvantages-of-automation.html. [Accessed 29 January 2013].

[2] S. Simpson, "Energy Efficiency the Next Big Trend in Building Construction and Renewal," The Vancouver Sun, 26 October 2012. [Online]. Available: http://www.vancouversun.com/business/bc2035/Energy+efficiency+next+trend+building+construction/7454460/story.html. [Accessed 29 January 2013].

[3] Made-in-China.com, "Hi-Speed I-Shape Tetra-Pack Straw Packing Machine," 30 January 2013. [Online]. Available: http://www.made-in-china.com/showroom/anna218/product-detailCoOEGsAlEbcI/China-Hi-Speed-I-Shape-Tetra-Pack-Straw-Packing-Machine-JH04-L-.html. [Accessed 30 January 2013].

[4] R. Emami, *Request for Proposal #3,* 2013.

[5] CoolMax, "I-500," 2012. [Online]. Available: http://www.coolmaxusa.com/productDetailsPower.asp?item=I-500&details=features&subcategory=120mm&category=single. [Accessed 20 January 2013].

[6] B. Ross, "Ben Ross' Blog," 22 September 2010. [Online]. Available: http://benross.net/wordpress/canada-trip-stop-4-ottawa/2010/09/22. [Accessed 29 January 2013].

[7] S. Ramsey, "Make a Candy Dispenser," 26 October 2012. [Online]. Available: http://www.woodworkingformeremortals.com/2012/10/make-candy-dispenser.html. [Accessed 29 January 2013].

[8] J. B. Merila, "Soda Straw Dispenser". United States of America Patent 4,227,626, 14 October 1980.

11.2 BIBLIOGRAPHY

"AVR LCD Thermometer Using ADC and PWM Project," ermicroblog, 3 January 2009. [Online]. Available:http://www.ermicro.com/blog/?p=519. [Accessed 23 March 2013].

"Microcontroller - A Beginners Guide - Button Debouncing Through Software," newbiehack, [Online]. Available: http://www.newbiehack.com/ButtonorSwitchDebounceinSoftware.aspx. [Accessed 25 March 2013].

"PIC18 Pulse Width Modulation (PWM) DC Motor Speed Controller with the RPM Counter Project," ermicroblog, 9 December 2009. [Online]. Available: http://www.ermicro.com/blog/?p=1461. [Accessed 1 April 2013].

APPENDICES

Appendix A - STANDARD OPERATING PROCEDURE

Pre-Operation

1. Load the light and dark dowels into their corresponding reservoirs; maximum 12 of each colour.
2. Load a maximum of 3 boxes into the box loader.
3. Follow the steps provided in the LCD-keypad user interface to provide the necessary operation information, including the number of boxes to be packed and the pattern corresponding to each box.
4. When all the information is confirmed, start the operation.

Operation

1. The conveyer belt motor is enabled and remains enabled until the operation ends or until the emergency stop is pressed.
2. The box loader rises to allow one box to fall onto the conveyer belt.
3. When the box arrives at the dowel dispenser, one dowel is pushed out at a time by the solenoid inside the reservoir and fall into the box.
4. When all 6 dowels are packed, the box continues to travel along the conveyer belt, and the guide rail supporting the lid will guide the lid to close.
5. At the end of the conveyer belt, the box will travel through the box-closing mechanism and land in the pick-up area. Refer to section 5.2.1 and Appendix B, figures 5-1.2 to 5-1.9 for a visual representation of the machine.
6. Steps 2 to 5 are repeated until all boxes are packed, at which time all actuators will be disabled.
7. Pressing the emergency stop button will halt the operation. Pressing it again will resume the operation.

Post-Operation

1. The following information will display on the LCD: Operation time, pattern corresponding to each packed box.
2. The system will return to Idle mode.

Appendix B - ADDITIONAL TABLES AND FIGURES

Figure 5-1.2: Position 1. Loading the box into the Box Loader

Figure 5-1.3: Position 2. Elevator has lifted the box to the loading ramp.

Figure 5-1.4: Position 3. Box slides down the loading ramp to assume position on the conveyor.

Figure 5-1.5: Position 4. Box is in position on the conveyor ready to move down the conveyor.

Figure 5-1.6: Position 5. Box is in position for dowel loading.

Figure 5-1.7: Position 6. Box is in position for the lid closer to shut the lid.

Figure 5-1.8: Position 7. Lid has been shut and is ready for the latch to lock it.

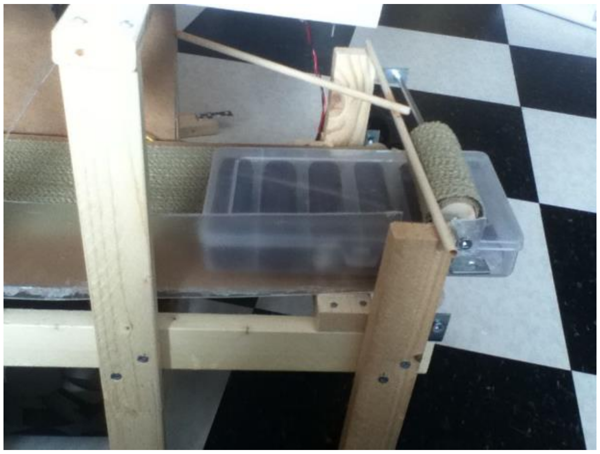


Figure 5-1.9: Position 8. Latch is being shut by the roller.

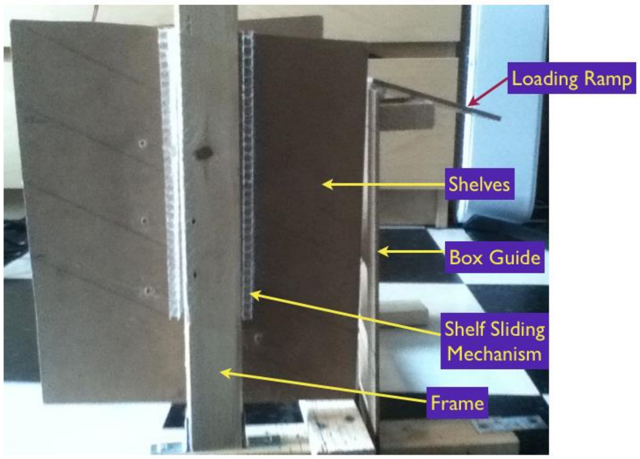


Figure 5-2.1: Side view of the box loader.

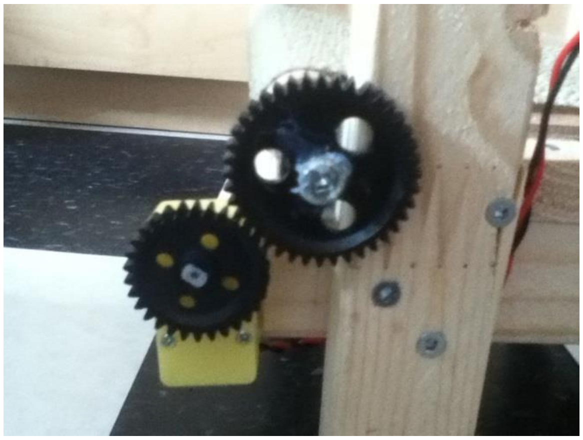


Figure 5-9.1: This shows the motor, a 12V Shenzhen DC Gearhead Motor, and gearbox, of ratio 4/3, connected to the dowel driving the conveyor belt.

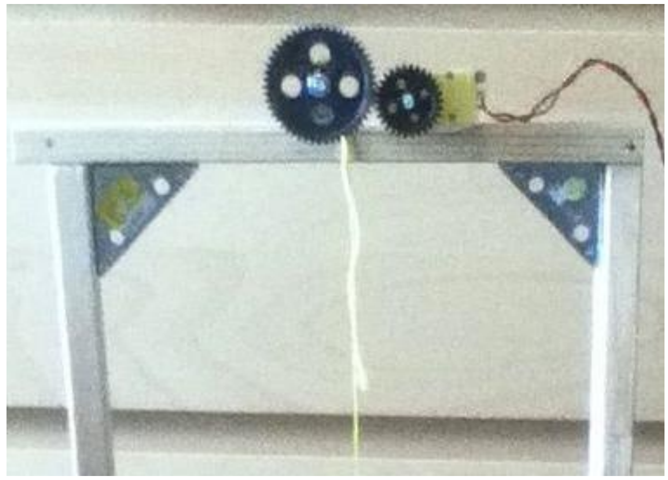


Figure 5-9.2: This shows the motor, a 12V Shenzhen DC Gearhead Motor, and gearbox, of ratio 5/3, connected to the box loader which drives the elevator.

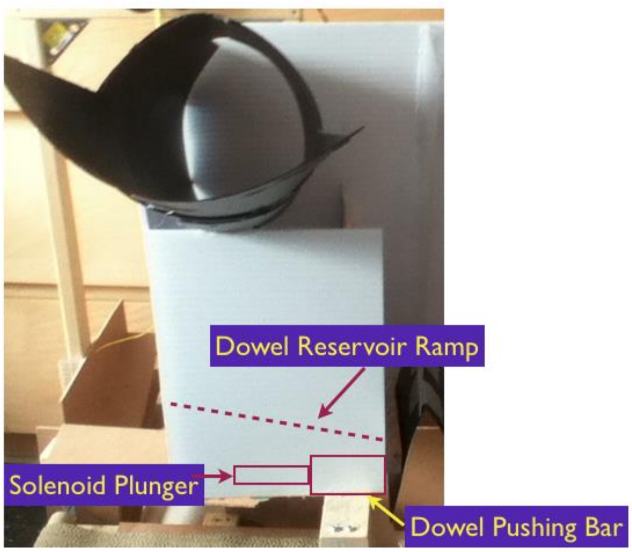


Figure 5-9.3: This figure shows a schematic of the placement of the dowel reservoir ramp, which angles the horizontally-aligned dowels, the solenoid plunger and the dowel pushing bar.

Figure 5-9.4: This figure shows a bird's eye view of the solenoid in the reservoir. This figure shows the placement of the left solenoid, the right solenoid is placed symmetrical to this one.

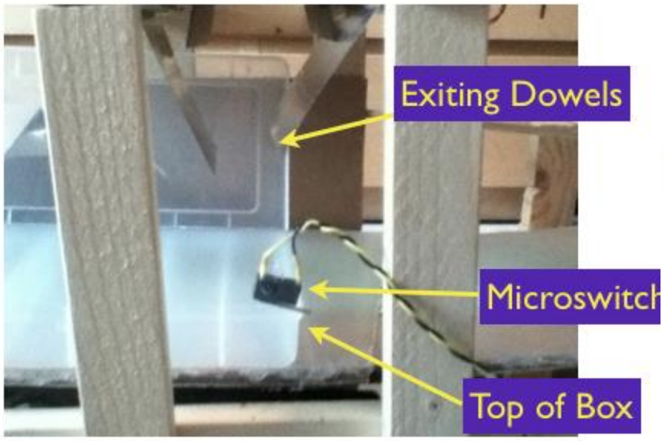


Figure 5-9.5: This shows the placement of the microswitch that detec when the empty box is in position, ready for dowel loading. The microswitch is placed just under the place where dowels exit the reservoir.

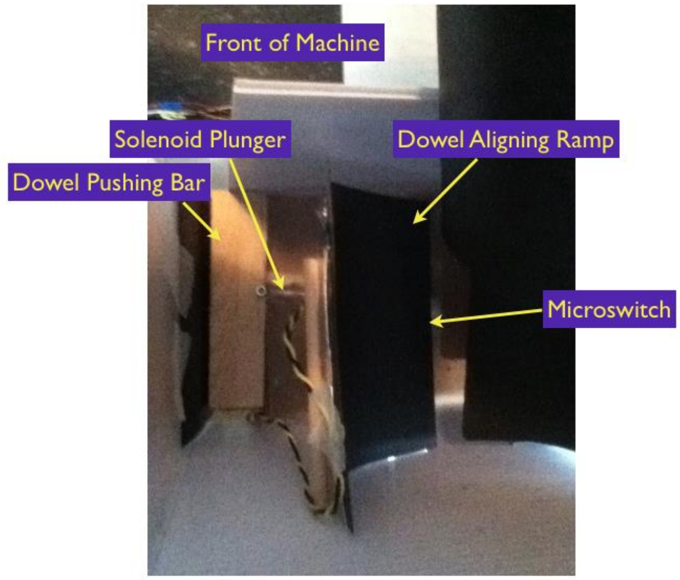


Figure 5-9.6: This shows the placement of the microswitch which counts the dowels. Horizontally aligned dowels roll down the dowel aligning ramp one at a time and fall down onto the dowel reservoir ramp.

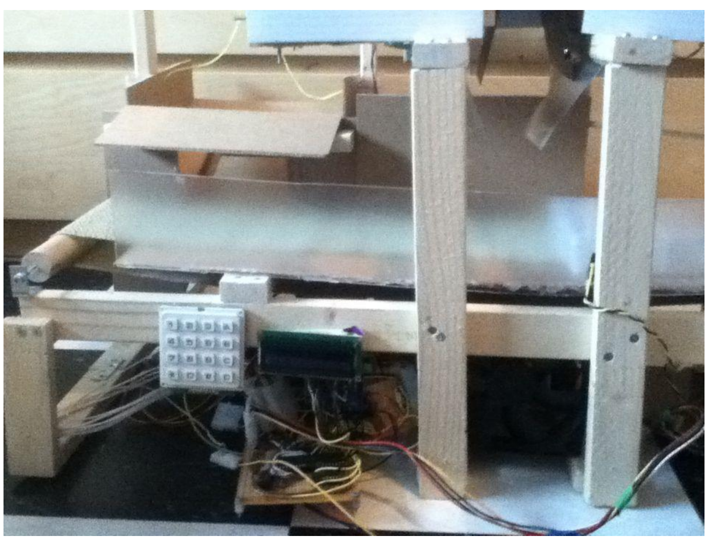


Figure 5-9.7: This figure shows the position of the LCD and Keypad. They are placed here, on the front of the machine so that they are accessible to the user, and clearly visible.

Figure 5-10.1: Loading Ramp: The loading ramp, situated just in front of the box loader, provides the sliding box exiting the box loader with enough momentum to position itself onto the conveyor as in Figure 5-10.2

Figure 5-10.2: This shows the effect of the loading ramp; the box is in position on the conveyor.

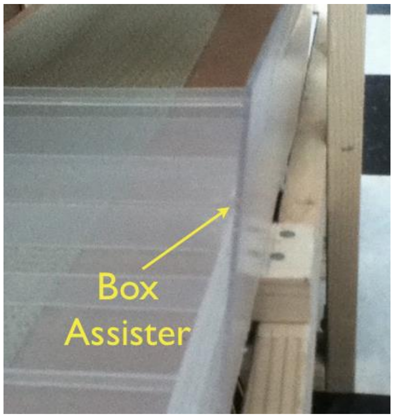


Figure 5-10.3: This shows a box assister which is a wall on the front side of the conveyor, which stops the box from sliding off the conveyor, ensuring it does not deviate from the position it must be in in order for a dowel to be loaded properly.

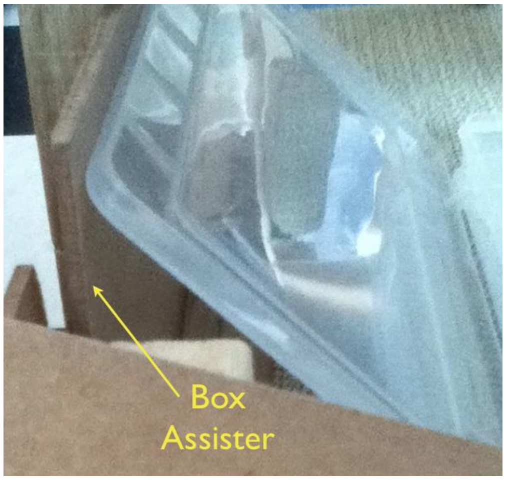


Figure 5-10.4: This is the third box assister. The box lid falls back on this once it has reached the dowel loading station. This ensures the lid does not interfere with dowel loading. The lid is raised so that the closing mechanism can shut it after loading.

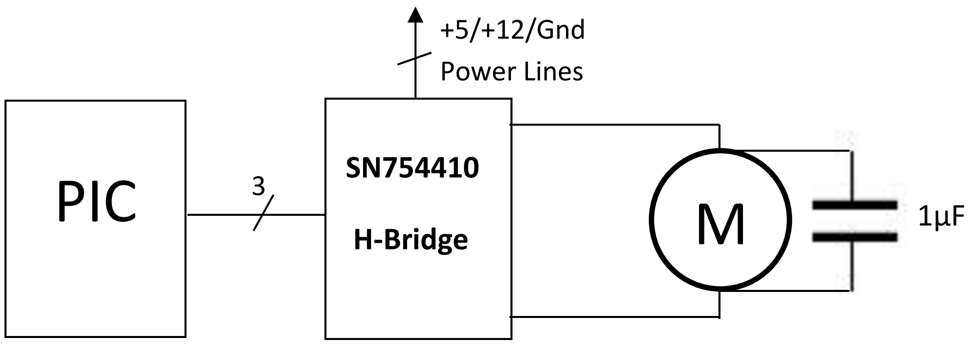
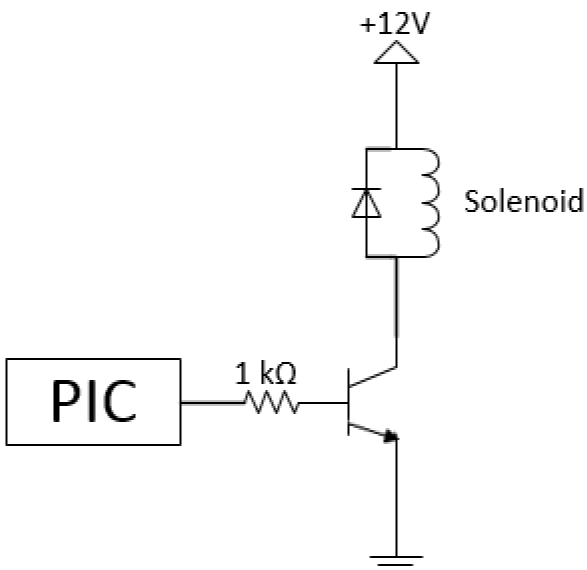


Figure 6-4: Schematic for the DC Motor. Refer to the relevant datasheet(s) for the pinout.



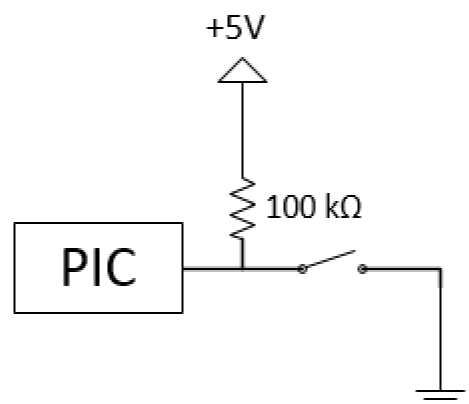


Figure 6-6: Schematic for the microswitches.

Figure 6-5: Schematic for the solenoids.

Table 7-1: Pin-by-pin description for the PIC18F4620 microcontroller

HD44780 (LCD)	
RS	RD2
R/W	GND
E	RD3
D4	RD4
D5	RD5
D6	RD6
D7	RD7
Solenoids	
Left	RD1
Right	RD2
Microswitches	
Right Reservoir	RB4
Left Reservoir	RB5
Conveyer Belt	RB2
MM74c922N (Keypad Encoder)	
Data Out A	RA0
Data Out B	RA1
Data Out C	RA2
Data Out D	RA3
Data Available	RB1
DC Motors	
Box Loader	
IN1	RC1
IN2	RC3
Conveyer Belt	
IN1	RC2
IN2	RC5
Emergency Stop Switch	
IN1	RB0

Appendix C - SOURCE CODE

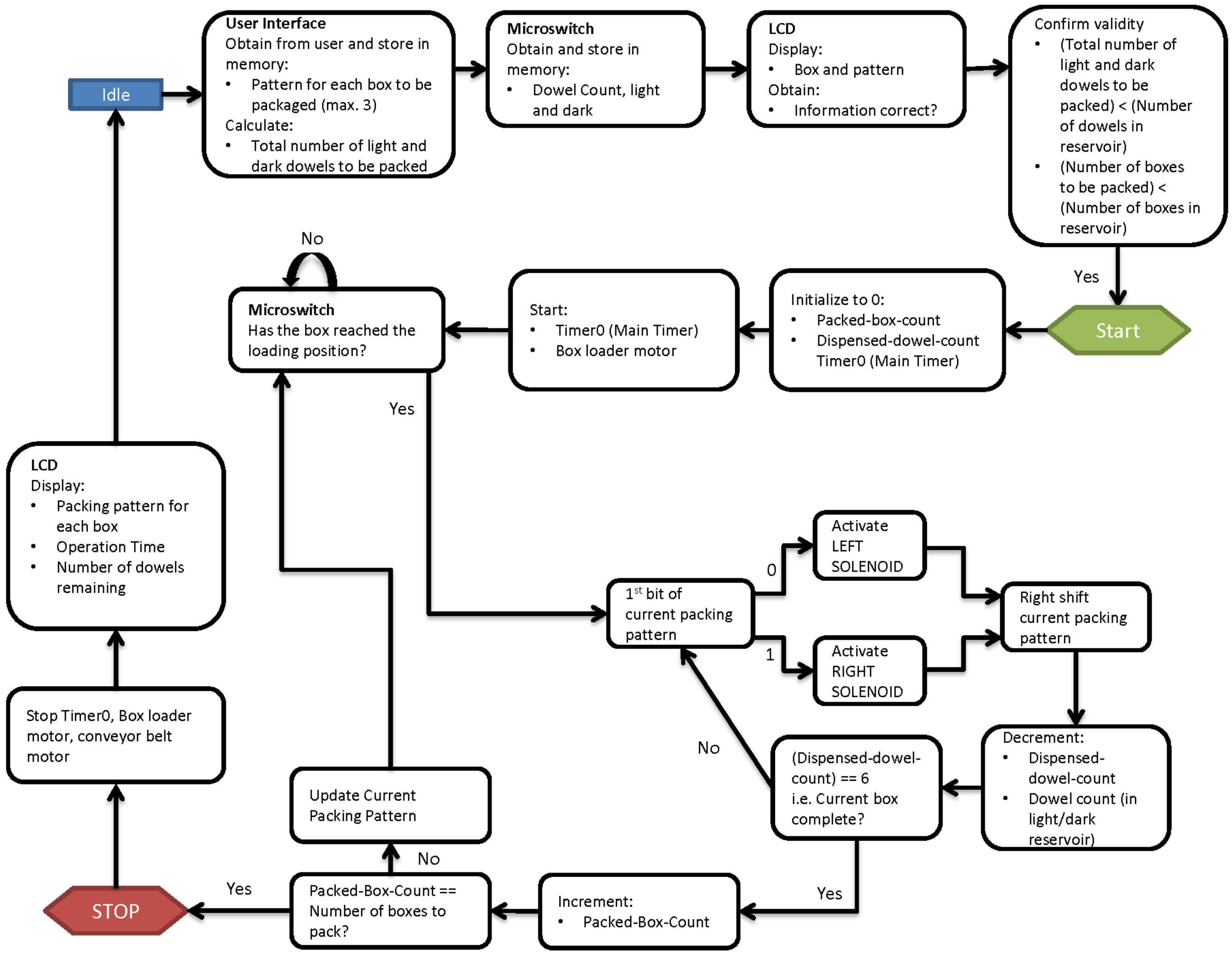


Figure C-0-1: Flowchart of the Main Program

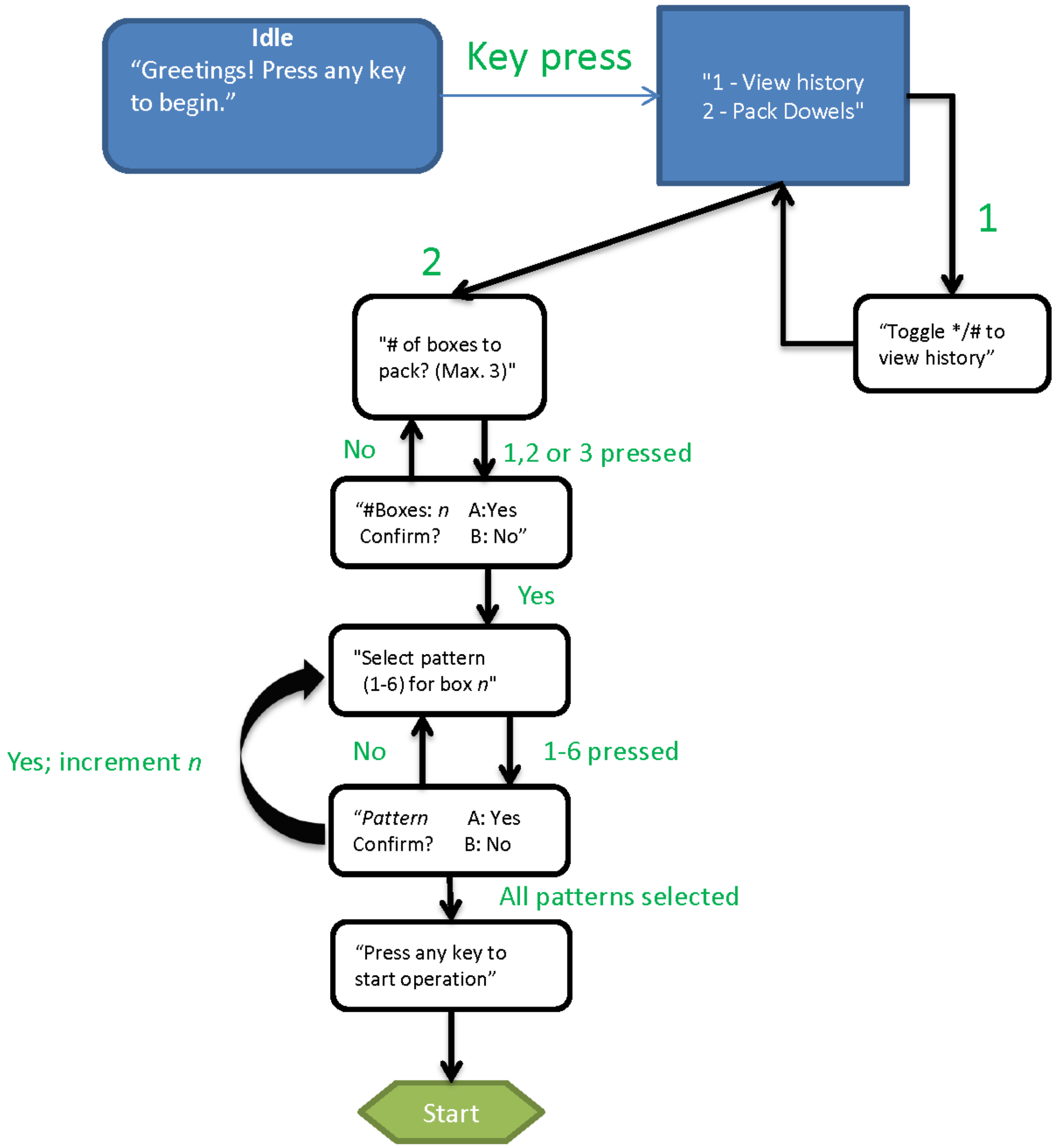


Figure C-0-2: User Interface Flow Chart

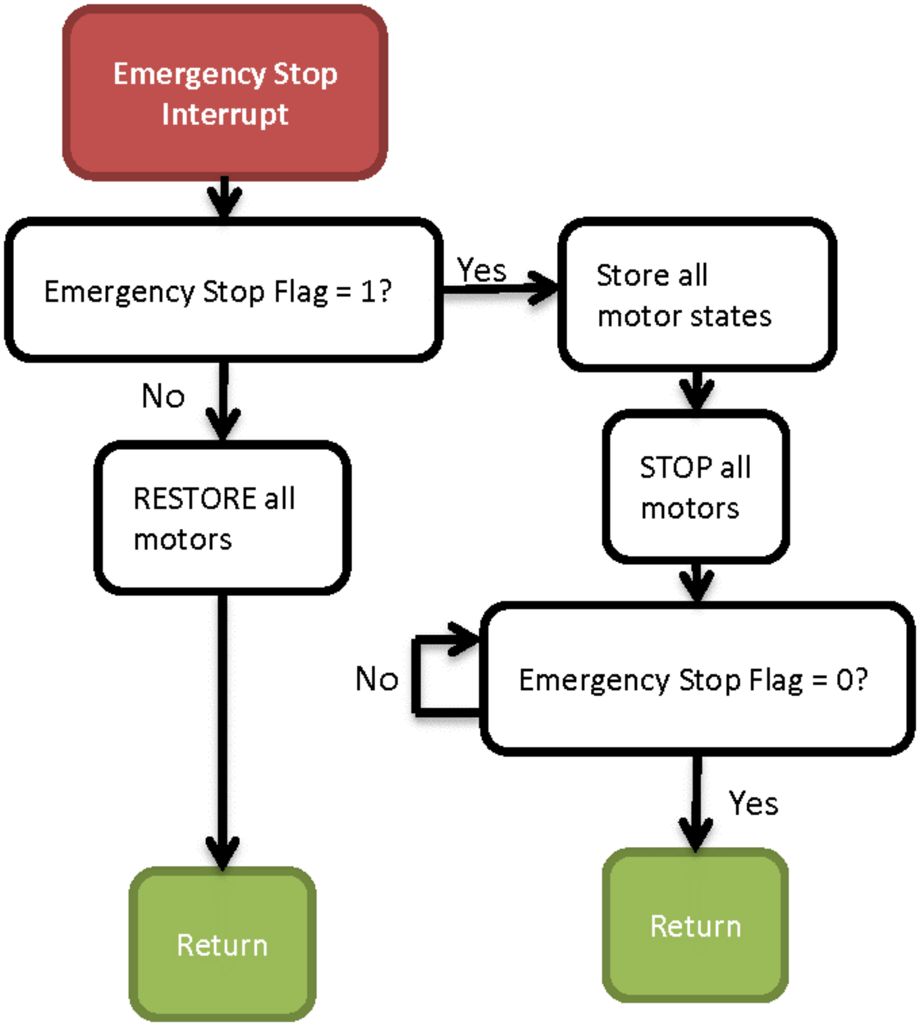


Figure C-0-3: Emergency Stop ISR Flow Chart

```
; P r o c e s s o r
;===============================================================================
      #include <p18f4620.inc>
        list P=18f4620, F=INHX32, C=160, N=80, ST=OFF, MM=OFF, R=DEC

;***************Configuration Bits**********************************************

             CONFIG OSC=INTIO67, FCMEN=OFF, IESO=OFF
             CONFIG PWRT = OFF, BOREN = SBORDIS, BORV = 3
             CONFIG WDT = OFF, WDTPS = 32768
             CONFIG MCLRE = ON, LPT1OSC = OFF, PBADEN = OFF, CCP2MX = PORTC
             CONFIG STVREN = ON, LVP = OFF, XINST = OFF
             CONFIG DEBUG = OFF
             CONFIG CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF
             CONFIG CPB = OFF, CPD = OFF
             CONFIG WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF
             CONFIG WRTB = OFF, WRTC = OFF, WRTD = OFF
             CONFIG EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF
             CONFIG EBTRB = OFF

; D a t a   M e m o r y
;===============================================================================

;Definitions

#define E                LATD, 3
#define RS               LATD, 2
#define LeftSolenoid     LATD, 1
#define RightSolenoid    LATD, 0
#define ConveyerMotor    LATC, 2
#define BoxLoaderMotor   LATC, 1
#define LeftMS           LATB, 5
#define RightMS          LATB, 4
#define ConveyerMS       LATC, 4

;Declare unbanked variables (at 0x70 and on)
        cblock 0x70
            Pattern_1
            Pattern_2
            Pattern_3
            currentPattern

;----LCD----
            temp_lcd
            dat
            delay1
            delay2
            Pattern_Counter
            TEMP1
            TEMP2
            TEMP3
            count
            Num
;----UI----
            Pattern
```

```
            box_count
            boxc_temp
            Confirmed_

;----PWM----
            d1
            d2
            d3

;----Switches----
            dowel_count_L
            dowel_count_R

            DispenseF   ;Dispense Flag

            EStopF
            ETemp_DispenseF
;----TIMER0----
            Sec_1
            Sec_ones
            Sec_tens
            Min_ones
            Min_tens

;----Operation----
            dowel_Dcount_L
            dowel_Dcount_R
            box_Dcount
            all_packedF
;----Temporary Variables----
            W_TEMP
            STATUS_TEMP
            BSR_TEMP

;----Debounce----
            Button
            Pressed
            Pressed_Clevel
            Released_Clevel

;----EEPROM----
            EE_Adr_current
            EE_Adr_start
        endc

; M A C R O S
;===============================================================================
CharToLCD macro char
; Display a keypad character on LCD
; Input: char - binary encoding of a character available on the keypad
; Return value: none
; Side effects: the character corresponding to 'char' is printed to the current
;   position on the LCD
        local end_
        movlw   upper KPHexToChar ;Load table pointer with full address of table
```

```
        movwf   TBLPTRU
        movlw   high KPHexToChar
        movwf   TBLPTRH
        movlw   low KPHexToChar
        movwf   TBLPTRL

        movf        char,W     ;Read char into W<3:0>
        andlw       0x0F
        addwf   TBLPTRL,f
        tblrd*              ;Read first Table entry into TABLAT
        movf    TABLAT, W                   ;Read Table Loop
        call        WR_DATA      ;Write the value in W to LCD
end_
        endm

History_keypad_poll_MACRO macro Go_Left, Go_Right
; For use in the 'View History' option; polls the keypad for either '*' or '#'
; Branches to 'Go_Left' if '*' is pressed, 'Go_Right' if '#' is pressed.
; Input: Go_Left, Go_Right (See description)
; Return Value: none
; Side Effects: Branches to the appropriate location.
        local       HKP_loop
        local       HKP_Cont
HKP_loop
        call        keypad_poll
        swapf       PORTA, W
        movwf       TEMP2

        movlw       b'00001100'
        cpfseq      TEMP2
        bra         HKP_Cont
        bra         Go_Left
HKP_Cont
        movlw       b'00001110'
        cpfseq      TEMP2
        bra         HKP_loop
        bra         Go_Right
        endm

Display macro Table_t
; Displays the string specified by Table_t onto the LCD
; Input: Table_t - a table containing the string to be printed.
; Return value: none
; Side effects: Table_t is printed onto the LCD

        local Again
        local end_
        movlw   upper Table_t ;Load table pointer with full address of table
        movwf   TBLPTRU
        movlw   high Table_t
        movwf   TBLPTRH
        movlw   low Table_t
        movwf   TBLPTRL

        tblrd*              ;Read first Table entry into TABLAT
```

```
        movf    TABLAT, W
Again                           ;Read Table Loop
        call    WR_DATA         ;Write to LCD
        tblrd+*                 ;Increment TBLPTR then read
        movf    TABLAT, W
        bnz     Again           ;Branch back to loop if Table does not return 0
end_
             endm

PUSH_MACRO MACRO
; For use in ISR's; Saves register contents
        movwf   W_TEMP
        movff   STATUS, STATUS_TEMP
        movff   BSR, BSR_TEMP
        ENDM

POP_MACRO MACRO
; For use in ISR's; Restores register contents
        movff   BSR_TEMP, BSR
        movff   STATUS_TEMP, STATUS
        movf    W_TEMP, W
        ENDM

Debounce_MACRO  macro Button_Reg, Button_Bit
; Debounces button. Button must be pressed for 50 ms to be considered 'pressed'.
; Input: Button_Reg; Button_Bit - Regisiter and bit to specify the button
; Return value: none
; Side effects: Pressed - set to 1 if the button is pressed, 0 if released
        local debounce_loop
        local Button_High
        local Button_Low
        local debounce_end
        clrf    Pressed_Clevel ; Pressed Confidence Level
        clrf    Released_Clevel ; Released Confidence Level
debounce_loop
        call    Delay_5ms
        btfss Button_Reg, Button_Bit
        bra   Button_Low
        bra   Button_High
Button_High
        clrf    Released_Clevel
        incf    Pressed_Clevel
        movlw   0x0A ; 10 x 5 ms
        cpfsgt  Pressed_Clevel
        bra    debounce_loop
        bsf    Pressed, 0
        bra     debounce_end
Button_Low
        clrf    Pressed_Clevel
        incf    Released_Clevel
        movlw   0x0A
        cpfsgt  Pressed_Clevel
        bra    debounce_loop
        bcf    Pressed, 0
debounce_end
```

```
        endm

EEPROM_WR macro EE_Adr, EE_Data
; Writes data stored at EE_Data to address specified by EE_Adr in the EEPROM
; Input: EE_Adr - 8-bit address; EE_Data.
; Return Value: none
; Side Effect: Data EE_Data is written to EE_Adr.
    clrf    EEADRH
    movf    EE_Adr, w
    movwf   EEADR
    movf    EE_Data, w
    movwf   EEDATA         ; Data Memory Value to write
    bcf     EECON1, EEPGD    ; Point to DATA memory
    bcf     EECON1, CFGS    ; Access EEPROM
    bsf     EECON1, WREN    ; Enable writes

    bcf     INTCON, GIE     ; Disable Interrupts
    movlw   0x55            ; Required Sequence
    movwf   EECON2          ; Required Sequence
    movlw   0xAA            ; Required Sequence
    movwf   EECON2          ; Required Sequence
    bsf     EECON1, WR      ; Required Sequence
    bsf     INTCON, GIE     ; Enable Interrupts
                            ; User code execution
    bcf     EECON1, WREN    ; Disable writes on write complete (EEIF set)
    endm

EEPROM_RD macro EE_Adr
; Reads data stored at EE_Adr of the EEPROM
; Input: EE_Adr
; Return Value: None
; Side effect: Data at EE_Adr is stored in WREG
    clrf    EEADRH
    movf    EE_Adr, w
    movwf   EEADR
    bcf     EECON1, EEPGD   ; Point to DATA Memory
    bcf     EECON1, CFGS    ; Access EEPROM
    bsf     EECON1, RD      ; EEPROM Read
    movf    EEDATA, W       ; W <- EEDATA
    endm
; R e s e t   V e c t o r
;===============================================================================
       org 0x0000

        goto initMAIN

; I n t e r r u p t   V e c t o r
;===============================================================================
        org 0x08        ; High-priority ISR
        call ISR_HIGH
        retfie

        org 0x18        ; Low-priority ISR
        call ISR_LOW
        retfie
```

```
;*************************************************
; Tables
;*************************************************
;TEST
Test                db "Hello world!", 0
;UI
Greeting
             db  "Greetings! Press any key to begin.",0
Welcome_Msg1
             db     "1 - View history   3 - Configure",0
Welcome_Msg2
             db "2 - Pack Dowels", 0
Box_sel1
                db "# of boxes to "0
Box_sel2
                db "pack? (Max. 3)",0
Box_number
                db "#Boxes:",0
Pattern_sel1
             db "Select pattern",0
Pattern_sel2
                db "(1-6) for box ",0
History
                db "Toggle to view history",0
Confirm
                db "   A:Yes",0
No
                db "Confirm?   B:No",0
Start_confirm
                db "Press any key to begin operation.", 0
Start_msg
                db "Starting yayyyy lalala~", 0
op_time
                db "Op.Time:",0
Pattern1
             db "1:000000",0
Pattern2
             db "2:111111",0
Pattern3
             db "3:000111",0
Pattern4
             db "4:001100",0
Pattern5
             db "5:110011",0
Pattern6
             db "6:010101",0
zero
                db "0",0
one
                db "1", 0
space_long      db "            ", 0

Operation_     db "Operation ",0
Box_Num_        db "# Boxes: "
```

```
Pattern_        db "Pattern ", 0

;Operation
space_          db " ",0
left
                db "Left: ",0
right
                db "Right: ",0
conveyer1
                db "Box in position",0
conveyer0
                db "Box complete", 0
now_packing     db "Packing: ",0

Pattern1_        db "Pattern 1:",0
Pattern2_        db "Pattern 2:",0
Pattern3_        db "Pattern 3:",0

Operation_end   db "Done Packing!",0
Operation_time  db "Operation Time: ",0
colon           db ":",0
EStop_Set
                db "EMERGENCY!!", 0
EStop_Clear
                db "All clear!      ",0

NumToChar       db "0123456789", 0

; M A I N
;===============================================================================

initMAIN
         movlw      b'11110010'     ; Set up PORTA for Keypad
         movwf      TRISA
         movlw     b'11111111'    ; Set required interrupt inputs
         movwf     TRISB
         clrf      TRISC          ; All port C is output
         clrf      TRISD          ; All port D is output

         clrf       FSR0H           ;Set FSR0 for accessing Pattern_X via
         movlw      0x70            ; indirect addressing
         movwf      FSR0L

         clrf       all_packedF     ; Clear all_packed flag
         clrf       box_Dcount      ; Clear counts
         clrf       box_count       ; Clear counts
         movlw      0x00            ; Set EEPROM address pointers
         movwf      EE_Adr_start    ; Set EEPROM address pointers
         movlw      0x00            ; Set EEPROM address pointers
         movwf      EE_Adr_current  ; Set EEPROM address pointers

         call   initLCD             ; Initialize LCD
         call   initPWM             ; Initialize PWM
         call   initSWITCHES        ; Initialize Micrswitches
         call   initTIMER0          ; Initialize TIMER0
```

```
         goto   MAIN

initLCD
; Initializes LCD
     call       lcdLongDelay
     call       lcdLongDelay
     movlw   b'00110011'
      call   WR_INS
      movlw  b'00110010'
      call   WR_INS

      ; 4 bits, 2 lines, 5x7 dots
      movlw  b'00101000'
      call   WR_INS

      ; display on/off
      movlw  b'00001111'
      call   WR_INS

      ; Entry mode
      movlw  b'00000110'
      call   WR_INS

      ; Clear ram
      movlw  b'00000001'
      call   WR_INS
       return

initSWITCHES
; Initializes microswitches
; Set variables to 0
        clrf        dowel_count_L
        clrf        dowel_count_R
        clrf        DispenseF

; Setup interrupts
        bcf         INTCON3, INT2IP     ; Set conveyer microswitch interrupt to low-
priorty
        bcf         INTCON,  RBIP       ; Set counter microswitch interrupts to low-
priorty
        bsf         INTCON3, INT2IE     ; Enable conveyer microswitch interrupt
        bsf         INTCON, INT0IE      ; Enable Emergency stop microswitch interrupt
        bsf         INTCON, RBIE        ; Enable couter microswitch interrupt
        bcf         INTCON3, INT2IF     ; Clear Conveyer interrupt flag
        bcf         INTCON, INT0IF      ; Clear Emergency stop interrupt flag
        bcf         INTCON, RBIF        ; Clear counter interupt flag
        bcf         INTCON2, INTEDG2    ; Conveyer belt switch interrupts on falling
edge
        bsf         INTCON2, RBPU       ; PortB Pull-up
        bsf         RCON, IPEN          ; Enable priority levels on interrupt
        bsf         INTCON, GIE         ; Global interrupt enable
        bsf         INTCON, PEIE        ; Peripheral interrupt enable
        return

initPWM
```

```
; Initialize PWM
        movlwB'01110000'  ;Set internal oscillator frequency to 8MHz
        movwfOSCCON
        clrf        TRISC        ;configure PORTC as output
        clrf    TRISD

; PWM Period = 4 x Tosc x (PR2 + 1) x TMR2 Prescale Value
; Tosc = 1/8 Mhz = 0.000000125
; PWM Period = 4 x 0.000000125 x 100 x 4 = 0.0002 seconds
; PWM Frequency = 1/PWM Period = 1/0.0002 = 5.0 kHz
        movlwB'01100011'  ;configure PR2
        movwfPR2
        movlw0x08   ;configure CCPR1L
        movwfCCPR1L
        movlw0x18   ;configure CCPR2L
        movwfCCPR2L
        movlwB'00000101'  ;configure T2CON, set prescaler to 4
        movwfT2CON
        return

initTIMER0
; Initialize TIMER0
        movlw   b'01000101' ; Set prescaler to 1:64
        movwf   T0CON
        movlw   b'00000110' ; Preload timer with 6
        movwf   TMR0L
        clrf    Sec_1
        clrf    Sec_ones
        clrf    Sec_tens
        clrf    Min_ones
        clrf    Min_tens
        bcf     INTCON, TMR0IF
        bsf     INTCON, TMR0IE
        return

; M A I N
;=============================================================================
MAIN
;=========================PRE-OPERATION======================================
UI_Main

        call        Clear_Display
        Display     Greeting
        call        ShiftDisplayLeft
menu
; "1 - View History   3 - Test Keypad
;  2 - Pack Dowels "
        call            Clear_Display
        Display         Welcome_Msg1
        call        Switch_Lines
        Display             Welcome_Msg2
        call            ShiftDisplayLeft

menu_next
        call keypad_poll
```

```
        movwf       TEMP1       ; Move value into TEMP1
        btfsc       TEMP1, 3    ; Check bits 3,2 and 1. Only proceed if they are all
0.
        goto        menu_next
        btfsc       TEMP1, 2
        goto        menu_next
        btfsc       TEMP1, 1
        goto        bit1_
        btfss       TEMP1, 0
        goto        history_init
        goto        pack_init
bit1_
        btfsc       TEMP1, 0
        goto        menu_next
        goto        test

history_init
        call        Clear_Display
        Display     History
history_loop0
        call        ShiftDisplayLeft
        swapf       PORTA, W
        andlw       0x0F
        movwf       TEMP2
        movlw       b'00001100'
        cpfseq      TEMP2
        bra         history_loop0
        bra         history
history
        call        Clear_Display
        Display     Operation_
history_OpTime
        call        Second_Line
        Display     op_time

        History_keypad_poll_MACRO  history_init, history_Pattern1

history_Pattern1
        call        Second_Line
        Display     Pattern_

        History_keypad_poll_MACRO  history_OpTime, history_Pattern2

history_Pattern2

        call        Second_Line
        Display     Pattern_
        History_keypad_poll_MACRO  history_Pattern1, history_Pattern3

history_Pattern3
        call        Second_Line
        Display     Pattern_
        History_keypad_poll_MACRO  history_Pattern2, history_init

        goto        menu
```

```
; Keypad test
test     call   Clear_Display
test_loop
         btfss            PORTA,1   ;Wait until data is available from the keypad
         goto       test_loop
         call           KeypadToLCD
         btfsc            PORTA,1     ;Wait until key is released
         goto       $-2
         goto       test_loop
KeypadToLCD
        movlw   upper KPHexToChar ;Load table pointer with full address of table
        movwf   TBLPTRU
        movlw   high KPHexToChar
        movwf   TBLPTRH
        movlw   low KPHexToChar
        movwf   TBLPTRL
        swapf       PORTA,W     ;Read PORTA<7:4> into W<3:0>
         andlw            0x0F
        addwf   TBLPTRL,f
        tblrd*              ;Read first Table entry into TABLAT
        movf    TABLAT, W                   ;Read Table Loop
         call       WR_DATA      ;Write the value in W to LCD
         return
KPHexToChar
          db        "123A456B789C*0#D"

pack_init
; "#of boxes to be packed"
        clrf        box_count
        call        Clear_Display
        Display     Box_sel1
        call        Switch_Lines
        Display     Box_sel2

box_num_loop
        call        keypad_poll
        btfsc       WREG, 3
        goto        box_num_loop
        btfsc       WREG, 2
        goto        box_num_loop
        btfss       WREG, 1
        goto        box_num_get
        btfss       WREG, 0
        goto        box_num_get
        goto        box_num_loop
box_num_get
; keypad input is 1-3 (valid); store number of boxes
        call        Clear_Display
        Display     Box_number
        swapf       PORTA, W
        movwf       box_count ; Temporarily store box count
        CharToLCD   box_count
        incf        box_count ; increment box_count by 1 to get actual value
        Display     Confirm
```

```
        call        Switch_Lines
        Display     No
        call        Confirm_AB
        btfsc       Confirmed_,0
        goto        pack_init
        clrf        boxc_temp ; to be compared to box_count
pack_loop
        movf        boxc_temp, W
        CPFSEQ      box_count
        goto        no_start
        goto        start
; " Select pattern
;   (1-6) for box _ "
no_start
        call        Clear_Display
        Display     Pattern_sel1
        call        Switch_Lines
        Display     Pattern_sel2
        CharToLCD   boxc_temp
;" <pattern>   A:yes
;   Confirm?   B:no "
poll_keypad
        call        keypad_poll
        swapf       PORTA,W     ;Read PORTA<7:4> into W<3:0>
        movwf       TEMP1       ; Move value into TEMP1
;Check bits to get selection
        btfsc       TEMP1, 3    ; invalid selection
        goto        poll_keypad
        btfss       TEMP1, 2
        goto        bitss0__
        goto        bitss1__
bitss1__
        btfsc       TEMP1, 1
        goto        bitss11_
        goto        bitss10_
bitss11_
        btfsc       TEMP1, 0
        goto        poll_keypad
        goto        bitss110

bitss10_
        btfss       TEMP1, 0
        goto        bitss100
        goto        bitss101
bitss0__
        btfsc       TEMP1, 1
        goto        bitss01_
        goto        bitss00_
bitss01_
        btfss       TEMP1, 0
        goto        bitss010
        goto        poll_keypad
bitss00_
        btfss       TEMP1, 0
        goto        bitss000
```

```
        goto        bitss001
bitss000
        call        Clear_Display
        Display     Pattern1
        movlw       b'000000'
        goto select_fin
bitss001
        call        Clear_Display
        Display     Pattern2
        movlw       b'111111'
        goto select_fin
bitss010
        call        Clear_Display
        Display     Pattern3
        movlw       b'000111'
        goto select_fin
bitss100
        call        Clear_Display
        Display     Pattern4
        movlw       b'001100'
        goto select_fin
bitss101
        call        Clear_Display
        Display     Pattern5
        movlw       b'110011'
        goto select_fin
bitss110
        call        Clear_Display
        Display     Pattern6
        movlw       b'010101'
        goto select_fin

select_fin
        movwf       TEMP2
        Display     Confirm
        call        Switch_Lines
        Display     No
; If A is pressed, store pattern, then return; if B is pressed, return.
        call        Confirm_AB
        btfsc       Confirmed_,0
        goto        pack_loop
        incf        boxc_temp
        movff       TEMP2, POSTINC1
        goto        pack_loop

start
        call        Clear_Display
        Display     Start_confirm
        call        ShiftDisplayLeft
        call        keypad_poll
        call        Clear_Display
        Display     Start_msg
        ;call        ShiftDisplayLeft
;========================OPERATION=============================================
MAIN_start
```

```
;;TestPWM
;     movlw   b'11111111'
;     movwf    LATC
;     call    ConveyerPWM_E
;     call    BoxLoaderPWM_E
;     bra     $
;TestSolenoid
;         bsf     LeftSolenoid
;         call    ShortDelay
;         call    ShortDelay
;         bcf     LeftSolenoid
;         call    ShortDelay
;         call    ShortDelay
;         bsf     RightSolenoid
;         call    ShortDelay
;         call    ShortDelay
;         bcf     RightSolenoid
;         call    ShortDelay
;         call    ShortDelay
;         bra     TestSolenoid
;TEST CASE
;     movlw   b'010101'
;     movwf   Pattern_1
;     movlw   b'100001'
;     movwf   Pattern_2
;     movlw   b'000111'
;     movwf   Pattern_3
;     movlw   0x02
;     movwf   box_count

    bsf     T0CON, TMR0ON ;Start Timer

;TMR_TEST
;     call    Clear_Display
;     movf    Min_tens, W
;     call    NumToLCD
;     movf    Min_ones, W
;     call    NumToLCD
;     Display colon
;     movf    Sec_tens, W
;     call    NumToLCD
;     movf    Sec_ones, W
;     call    NumToLCD
;     call    ShortDelay
;     bra     TMR_TEST

    call    ConveyerPWM_E

MAIN_loop
    call    box_count_Check
    btfsc   all_packedF, 0
    bra     MAIN_end
    call    Load_box
    call    currentPattern_Update
```

```
check
    btfss   DispenseF, 0        ; Check flag; if 0, do not dispense
    bra     check

    call    Dispense
    incf    box_Dcount
    bra     MAIN_loop

MAIN_end
    bcf     T0CON, TMR0ON ;Stop Timer
    call    Delay           ; Move the last box to the end of conveyer before
    call    ConveyerPWM_D   ; turning off

;=============================POST-OPERATION====================================
;Store data in EEPROM
    EEPROM_WR   EE_Adr_current,box_count
    incf        EE_Adr_current
    EEPROM_WR   EE_Adr_current, Min_tens
    incf        EE_Adr_current
    EEPROM_WR   EE_Adr_current, Min_ones
    incf        EE_Adr_current
    EEPROM_WR   EE_Adr_current, Sec_tens
    incf        EE_Adr_current
    EEPROM_WR   EE_Adr_current, Sec_ones
    incf        EE_Adr_current


    EEPROM_WR   EE_Adr_current,Pattern1
    incf        EE_Adr_current

    EEPROM_WR   EE_Adr_current,Pattern2
    incf        EE_Adr_current

    EEPROM_WR   EE_Adr_current,Pattern3
    incf        EE_Adr_current
EEPROM_end
;Display data
    call    Clear_Display
    Display Operation_end
    call    Switch_Lines

; Display Operation Data on LCD
Display_loop
    Display Operation_time
    call    Delay
    call    Second_Line
    movf    Min_tens, W
    call    NumToLCD
    movf    Min_ones, W
    call    NumToLCD
    Display colon
    movf    Sec_tens, W
    call    NumToLCD
    movf    Sec_ones, W
    call    NumToLCD
```

```
    Display space_long
    call    Delay

    call    Second_Line
    movff   box_count, TEMP2
    Display Pattern1_
    movf    Pattern_1,W
    call    PatternToLCD
    call    Delay
    call    Second_Line
    dcfsnz  TEMP2
    bra     Display_loop

    Display Pattern2_
    movf    Pattern_2,W
    call    PatternToLCD
    call    Delay
    call    Second_Line
    dcfsnz  TEMP2
    bra     Display_loop

    Display Pattern3_
    movf    Pattern_3,W
    call    PatternToLCD
    call    Delay
    call    Second_Line
    bra     Display_loop

    goto $

;***************************************
; UI Subroutines
;***************************************
Confirm_AB
; Polls keypad for user input.
; If A is pressed, Confirmed_ = 0;
; If B is pressed, Confirmed_ = 1;
        movlw       b'1'
        movwf       Confirmed_
confirm_loop
        call        keypad_poll
        btfsc       WREG, 3
        goto        confirm_loop
        btfss       WREG, 1
        goto        confirm_loop
        btfss       WREG, 0
        goto        confirm_loop
        btfss       WREG, 2
        goto        A_pressed
        goto        B_pressed
A_pressed
        clrf       Confirmed_
        return
B_pressed
        return
```

```
keypad_poll
; Polls keypad for input. Store pressed key in WREG.
         btfss           PORTA,1     ;Wait until data is available from the keypad
         goto      $-2
         clrf       WREG
         swapf           PORTA,w     ;Read PORTA<7:4> into W<3:0>
         andlw           0x0F
         btfsc           PORTA,1     ;Wait until key is released
         goto      $-2
         return

;***************************************
; Operation Subroutines
;***************************************

Dispense
; Dispenses 6 dowels into the box according the
;   pattern specified by currentPattern
        movlw   0x06
        movwf   count
Dispense_again
        btfss   DispenseF, 0        ; Check flag; if 0, do not dispense
        return
        btfsc   currentPattern, 0
        bra     dispenseLeft
        bra     dispenseRight

dispenseLeft
        decf    dowel_count_L
        incf    dowel_Dcount_L
        bsf     LeftSolenoid
        call    Delay
        bcf     LeftSolenoid
        call    ShortDelay
        bra     Dispense_cont
dispenseRight
        decf    dowel_count_R
        incf    dowel_Dcount_R
        bsf     RightSolenoid
        call    Delay
        bcf     RightSolenoid
        call    ShortDelay
Dispense_cont
        rrncf   currentPattern
        decfsz   count
        bra    Dispense_again
        return

currentPattern_Update
        movff   POSTINC0, currentPattern
        return

box_count_Check
```

```
        movf    box_Dcount, W
        CPFSEQ  box_count
        return
        bsf     all_packedF, 0
        return

Load_box
        call    BoxLoaderPWM_E
        call    Delay
        call    BoxLoaderPWM_D
        return

;****************************************
; ISRs
;****************************************
ISR_HIGH
; High priority interrupt handler
        PUSH_MACRO ; Save registers
        btfsc   INTCON, INT0IF
        call    ISR_EStop
        POP_MACRO ; Restore registers
        return

ISR_LOW
; Low priority interrupt handler
        PUSH_MACRO ; Save registers
        btfsc   INTCON, TMR0IF
        call    ISR_Timer0
        btfsc   INTCON, RBIF
        call    ISR_PortB
        btfsc   INTCON3, INT2IF
        call    ISR_Conveyer
        POP_MACRO ; Restore registers
        return

ISR_Timer0
; TIMER0 overflow interrupt handler;
; Input: -
; Output: -
; Side Effects: Increments time recoreded by
;       Sec_1, Sec_ones, Sec_tens, Min_ones and Min_tens
    bcf     INTCON, TMR0IF ; Clear interrupt flag
    movlw   b'00000110' ; Preload timer with 8
    movwf   TMR0L
    bsf     INTCON, TMR0IE ; Re-enable interrupt
    incf    Sec_1
    movlw   0x7d  ; Increment Sec_1 every 125 TIMER0 overflows
    cpfseq  Sec_1
    return
    bra     Inc_Sec_ones
Inc_Sec_ones
    clrf    Sec_1
    incf    Sec_ones
    movlw   b'00001010'
    subwf   Sec_ones, W
```

```
    btfsc   STATUS, Z
    bra     Inc_Sec_tens
    return
Inc_Sec_tens
    clrf    Sec_ones
    incf    Sec_tens
    movlw   b'00000110'
    subwf   Sec_tens, W
    btfsc   STATUS, Z
    bra     Inc_Min_ones
    return
Inc_Min_ones
    clrf    Sec_tens
    incf    Min_ones
    movlw   b'00001010'
    subwf   Min_ones, W
    btfsc   STATUS, Z
    bra     Inc_Min_tens
    return
Inc_Min_tens
    clrf    Min_ones
    incf    Min_tens
    return

ISR_PortB
; PortB Change interrupt handler;
; Input: -
; Output: -
; Side Effects: calls ISR corresponding to appropriate PORTB bit
    movf    PORTB, W
    bcf     INTCON, RBIF     ; Must read > clear flag > read to avoid mismatch
    movf    PORTB, W
;    btfss   PORTB, 1       ; Uncomment if key 'D' is used as the emergency stop
;    bra     ISR_Keypad
    btfss   PORTB, 4         ; Check PortB Pins
    bra     ISR_dowel_count_R
    btfss   PORTB, 5
    bra     ISR_dowel_count_L
    return

ISR_dowel_count_R
; Counter Microswitch interrupt handler
; Increments the count of the right reservoir; displays count to LCD
    Debounce_MACRO PORTB, 4
    btfss   Pressed,0
    return
    bcf     INTCON, RBIF
    call    Clear_Display
    Display right
    incf    dowel_count_R
    movf    dowel_count_R, W
    call    PatternToLCD
    bsf     LATB, 4
    bcf     INTCON, RBIF
    return
```

```
ISR_dowel_count_L
; Counter Microswitch interrupt handler
; Increments the count of the left reservoir; displays count to LCD
    Debounce_MACRO PORTB, 5
    btfss   Pressed,0
    return
    bcf     INTCON, RBIF
    call    Clear_Display
    Display left
    incf    dowel_count_L
    movf    dowel_count_L, W
    call    PatternToLCD
    bsf     LATB, 5
    bcf     INTCON, RBIF
    return

;ISR_Keypad
;; Uncomment if 'D' key is used as the Emergency Stop
;    bcf     INTCON, RBIF
;    clrf           WREG
;     swapf         PORTA,w     ;Read PORTA<7:4> into W<3:0>
;     andlw         0x0F
;     movwf      TEMP3
;     movlw      b'1111'
;    cpfseq      TEMP3
;    return
;    call        ISR_EStop
;    return

ISR_Conveyer
; Conveyer belt microswitch interrupt handler
;
    Debounce_MACRO PORTB, 2
    btfss   Pressed,1
    return
    bcf     INTCON3, INT2IF
    btg     INTCON2, INTEDG2 ; Invert edge-select
    btg     DispenseF, 0     ; Toggle Dispense Flag
    call    Clear_Display
    btfss   DispenseF, 0
    bra     Dispense0
    bra     Dispense1
Dispense0
    Display conveyer0
    return
Dispense1
    Display conveyer1
    call    Switch_Lines
    Display now_packing
    movf    currentPattern, W
    call    PatternToLCD
    return

ISR_EStop
; Emergency stop interrupt handler
```

```
    Debounce_MACRO PORTB, 0
    btfss   Pressed,0
    return
    bcf     INTCON, INT0IF
    bsf     INTCON, GIE
    call    Clear_Display
    btg     EStopF, 0
    btfss   EStopF, 0
    bra     EStop0
    bra     EStop1
EStop0
    movff   ETemp_DispenseF, DispenseF  ; Restore Dispense flag
    btfss   all_packedF, 0              ; If not all packed, start conveyer
    call    ConveyerPWM_E
    Display EStop_Clear
    return
EStop1
    movff   DispenseF, ETemp_DispenseF
    call    ConveyerPWM_D ;Stop all motors
    call    BoxLoaderPWM_D
    bcf     LeftSolenoid
    bcf     RightSolenoid

    Display EStop_Set
EStop_poll
    btfss   EStopF, 0
    bra     EStop0
    bra     EStop_poll

;***************************************
; LCD control
;****************************************
PatternToLCD
;Display a pattern to the LCD
;Input: Pattern specified in a 6-bit binary value
;Output: -
;Side effect: Displays pattern to LCD in binary
        movwf   TEMP3
        movlw   0x06
        movwf   count
Again2
        btfsc   TEMP3, 5
        goto one_
        goto zero_
one_
        Display one
        goto cont
zero_
        Display zero
cont
        rlncf   TEMP3
        decfsz   count
        goto    Again2
        return
```

```
NumToLCD
; Displays digit (0-9) to the LCD
; Input: Num
; Output: -
; Side effects: the digit 0-9 is displayed to the current position on the LCD
        movwf   Num
        movlw   upper NumToChar ;Load table pointer with full address of table
        movwf   TBLPTRU
        movlw   high NumToChar
        movwf   TBLPTRH
        movlw   low NumToChar
        movwf   TBLPTRL

        movf       Num,W     ;Read number into W<3:0>
        andlw      0x0F
        addwf   TBLPTRL,f
        tblrd*              ;Read first Table entry into TABLAT
        movf    TABLAT, W                   ;Read Table Loop
        call       WR_DATA      ;Write the value in W to LCD
        return

Switch_Lines
; Moves cursor to the other line
             movlw  B'11000000'
             call   WR_INS
             return

ShiftDisplayLeft
; Shifts display to the left until a key is pressed
        movlw      b'00011000'        ;Move to the left
             call         WR_INS
             call         lcdLongDelay
        btfss      PORTB,1     ;Check if data is available from the keypad
             goto         ShiftDisplayLeft                 ;repeat operation
                return

Second_Line
; Moves cursor to the second line
                movlw   B'00000010'
                call    WR_INS
                call    Switch_Lines
                return

Clear_Display
; Clears LCD
             movlw  B'00000001'
             call   WR_INS
             return

WR_INS
;Write command to LCD
;Input: W
;Output: -
    bcf     RS                           ;clear RS
    movwf    temp_lcd                 ;store instruction
```

```
      andlw  0xF0               ;mask 4 bits MSB w = X0
      movwf  LATD               ;Send 4 bits MSB
      bsf    E                        ;
    call     lcdLongDelay ;__    __
      bcf          E                       ;  |__|
    swapf   temp_lcd, W
    andlw   0xF0
      movwf  LATD               ;send 4 bits LSB
      bsf          E                       ;
      call   lcdLongDelay ;__    __
      bcf          E                       ;  |__|
      call   lcdLongDelay
      return

WR_DATA
;Write data to LCD
;Input: W
;Output: -
      bsf     RS
      movwf  dat
      movf   dat, W
      andlw  0xF0
      addlw  4
      movwf  PORTD
      bsf     E
      call   LCD_DELAY       ;__    __
      bcf     E                   ;  |__|
      swapf  dat, W
      andlw  0xF0
      addlw  4
      movwf  PORTD
      bsf     E                   ;
      call   LCD_DELAY       ;__    __
      bcf     E               ;  |__|
    call    LCD_DELAY
      return

;****************************************************
; PWM Controls
;****************************************************
BoxLoaderPWM_E
; Enables Box Loader Motor
                bsf CCP1CON, CCP1M3
                bsf     CCP1CON, CCP1M2
                bsf     LATC, 3
                return
BoxLoaderPWM_D
; Disables Box Loader Motor
                bcf CCP1CON, CCP1M3
                bcf     CCP1CON, CCP1M2
                bcf     LATC, 3
                return

ConveyerPWM_E
; Enables Conveyer Belt Motor
```

```
                bsf     CCP2CON, CCP2M3
                bsf     CCP2CON, CCP2M2
                bsf     LATC, 4
                return
ConveyerPWM_D
; Disables Conveyer Belt Motor
                bcf     CCP2CON, CCP2M3
                bcf     CCP2CON, CCP2M2
                bcf     LATC, 4
                return

;****************************************************
; Delays
;****************************************************

    ;Delay: ~5s
Delay
                    ;9999995 cycles
      movlw  0x5A
      movwf  d1
      movlw  0xCD
      movwf  d2
      movlw  0x16
      movwf  d3
Delay_0
      decfsz d1, f
      goto   $+6
      decfsz d2, f
      goto   $+6
      decfsz d3, f
      goto   Delay_0

                    ;3 cycles
      goto   $+4
      nop
                    ;4 cycles (including call)
      return

; Delay = 0.25 seconds
; Clock frequency = 32 MHz

; Actual delay = 0.25 seconds = 2000000 cycles
; Error = 0 %
ShortDelay      ;1999996 cycles
      movlw  0x11
      movwf  d1
      movlw  0x5D
      movwf  d2
      movlw  0x05
      movwf  d3
ShortDelay_0
      decfsz d1, f
      goto   $+6
      decfsz d2, f
      goto   $+6
```

```
        decfsz d3, f
        goto   ShortDelay_0

                     ;4 cycles
        goto   $+2
        goto   $+2
        return
;Delay: ~44us
LCD_DELAY
        movlw   0x33
        movwf   delay1, 0

Delay44usLoop
        decfsz  delay1, f
        bnz     Delay44usLoop
        return

    ;Delay: ~1ms
lcdLongDelay
    movlw 0x8f
    movwf delay1
    movwf 0x02
    movwf delay2, 0

LLD_LOOP
    decfsz delay1,f
    bra de2
    decfsz delay2, f
de2  bra LLD_LOOP
    return

; Delay = 0.005 seconds
; Clock frequency = 32 MHz

; Actual delay = 0.005 seconds = 40000 cycles
; Error = 0 %

Delay_5ms
                     ;39998 cycles
        movlw  0x3F
        movwf  d1
        movlw  0x20
        movwf  d2
Delay_1
        decfsz d1, f
        goto   $+6
        decfsz d2, f
        goto   Delay_1

                     ;2 cycles
        goto   $+2
         return
    END
```

Appendix D - REQUEST FOR PROPOSAL

AER201 - Engineering Design — *Winter 2013*

Request for Proposal #3

The Dowel-packer Machine

Need

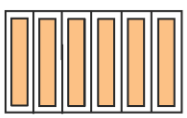
A utility company needs to package two colours of wood dowels in organizer boxes in various mixed combinations.

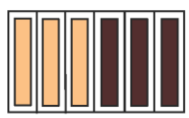
Goal

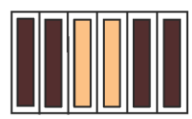
Design and manufacture the proof-of-concept prototype of a machine that can package light and dark wood dowels in organizer boxes in a variety of combinations reliably and quickly, according to the operator's keyed-in requests.

Specifications

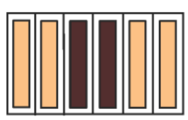
The machine is expected to package 2 boxes, in each 6 wood dowels, in no longer than 2 minutes. Each dowel is cylindrical with a diameter of $25.5^{\pm 0.5}$mm, a height of $105^{\pm 2}$mm and a weight of $25^{\pm 2}$g. The dowels are to be supplied in no particular format into two separate reservoirs in the machine, one for light and one for dark dowels. A maximum number of 12 dowels may be supplied to each reservoir, but their exact number is not known *a priori*. The boxes are made of frosted plastic with a hinged cover with a latch in the middle that snaps when it is closed. The overall dimensions of each box are $209^{\pm 2}$mm (length) by $121^{\pm 2}$mm (width) by $35^{\pm 2}$mm (height). There are 6 compartments in the box, each with the dimensions of $115^{\pm 2}$mm by $32^{\pm 2}$mm by $30^{\pm 2}$mm. The weight of each empty box is $115^{\pm 5}$g. Samples of boxes and dowels are available from the client for a closer examination. A minimum number of 3 boxes should be supplied to the machine for each operation. Methods of loading and delivering dowels and boxes are up to the design, but must be convenient for the operator (e.g., easily accessible and no need for disassembling any parts.) In each box, a combination of light and dark dowels must be placed in the compartments in a pattern specified by the operator through a keypad in the beginning of the operation. Possible patterns are shown below, and there will be two different patterns for the two boxes to be packaged in each operation.

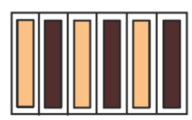
Pattern #1:


Pattern #3:


Pattern #5:


Pattern #2:


Pattern #4:


Pattern #6:


After placing all the dowels in each box, it must be placed in the pickup location/bin with its cover completely closed (snapped). After each operation, the machine is expected to return to the standby mode, display a completion or termination message on the LCD, and be ready to communicate with the operator the operation information. Also, all the remaining dowels must be returned in their reservoir after each operation. The information to be retrieved from the machine after each operation shall include: operation time, number of dowels remaining in each reservoir, and the packaging pattern for each box. The machine shall accept operator's instructions for packaging options through a keypad. The menus displayed on the LCD must be self-explanatory and provide easy navigation for operators of various skill levels. The client requires that the machine be portable with no need for installations, and as such there are constraints on weight and dimensions. Also for safety purposes the machine must have an easily-accessible emergency STOP switch that stops all the mechanical moving parts immediately. The machine can be plugged into the AC outlet.

Operation

The machine is normally in the standby mode. After loading the dowels and boxes, the operator enters on a keypad the instructions about the packaging patterns, and then starts the operation by pressing a *<start>* button on the keypad. The entire packaging process must be done autonomously, and must take no longer than 2 minutes. Upon completion of the operation, all the remaining dowels must be returned in their reservoir, and packed boxes be placed in the pickup area/bin. Then, the machine returns to the standby mode by displaying a completion or termination message on the display. The operator can then communicate with the machine through the keypad and display to retrieve the operation information.

Machine performance will be evaluated depending on the operation time, the quality of packaging (dowels placed properly, covers closed), and the accuracy of the retrieved information, as detailed in the sequel.

Performance Evaluation

The prototype will run two separate but consecutive operations, and the total time, quality, and accuracy of these operations are measured. Reward and Penalty points will be given to the prototype performance according to the following scheme. Each operation is qualified for scoring if the machine delivers at least one packed and closed box (with minimum 5 dowels in it, even if not placed or sorted correctly) to the pickup location/bin within 2 minutes, returns to the standby mode by displaying the completion or termination message at the end of its operation, and prompts for the packaging information.

➢ Each qualified operation	+1000
➢ Each delivered box "packaged correctly"	+500 / box
➢ Each delivered box "closed completely	+500 / box
➢ Missing dowel in the boxes	-300 / dowel
➢ "Damaged" dowel	-500 / dowel
➢ Dowel placed in the correct compartment	+200 / dowel
➢ Box not delivered to the pickup location/bin	-300 / box
➢ Remaining dowels not returned to their reservoir at the end of operation	-500 / reservoir
➢ The recorded pattern for the box shown on the display is incorrect	-200 / box
➢ The displayed number of remaining dowels in each reservoir shown on the display is incorrect	-500 / reservoir
➢ The operation time recorded on the display is "incorrect"	-500
➢ Time penalty:	- 10 per second of operation
➢ Each disqualified operation	0
Bonus Points for Extra Features:	
➢ Robustness and Durability	0 to -300
➢ Operability and Sustainability	0 to · 300

- Elegance and Safety 0 to +300
- Extendibility 0 to +300
- Dexterity 0 to +600
- Compactness and Portability −500
- Real-time Date/Time Display −400
- Permanent Logs −400
- PC Interface −300

Constraints

a. The entire prototype, excluding the packed boxes, pickup bin, and extended cable to the AC plug, shall completely fit within a $75\times75\times75$ cm^3 envelope at all operation times.

b. The weight of the machine (without the dowels and boxes) shall not exceed 6 kg.

c. The total prototype costs shall not exceed \$230CDN.

d. The machine can be plugged in the AC, 110V-60Hz, 3-pin outlet or use its own on-board power supply during the operation.

e. The machine must have an easily-accessible emergency STOP switch that stops all the mechanical moving parts immediately.

f. The machine must be fully autonomous, and no interaction with an external PC or remote control is permitted during the operation. The operation must start by pressing a *<start>* button on the keypad.

g. No installation or instrumentation is allowed in addition to what is devised within the machine.

h. The locations for supplying dowels and boxes and also the pickup location/bin must be specified in the machine clearly.

i. Dispensing the packed boxes and retrieving the returning dowels must be easy with no need for disassembling any parts of the machine.

j. At the end of each run, the machine display must be on prompt to show the following information per operator's request: operation time, number of dowels remaining in each reservoir, and the packaging pattern for each box.

k. The machine user interface for both operation and information retrieval shall be self-explanatory, and provide easy navigation for operators of various skill levels.

l. Each box is "packaged correctly" if all dowels are placed according to the operator's selected pattern for the particular box.

m. Each box is "closed completely" only if the cover is completely closed and the latch is snapped.

n. Each box or dowel (in the box, reservoir or machine) is considered as "damaged" if there are clear defects as a result of the operation, to the referee's discretion.

o. The operation time is the duration between when the *<start>* button on the keypad is pressed and when the termination/completion message is shown on the machine LCD. The operation time shall not exceed 2 minutes. Further, the time required for entering the operator's instructions on the keypad before the operation shall not exceed 1 minute.

p. The recorded operation time is considered "correct" if it is equal to the time measured by the referee ± 5%. Otherwise, it is assumed "incorrect."

q. The machine's response to the operator's inquiry about the packaging statistics is considered "correct" if it is according to the outcome of the performed operation, otherwise it is considered as "incorrect."

r. Each operation is "qualified" for scoring if the machine delivers at least one packed and closed box (with minimum 5 dowels in each, even if not placed or sorted correctly) to the pickup location/bin within 2 minutes, returns to the standby mode by displaying the completion or termination message at the end of its operation, and prompts for the packaging information.

s. Each operation is "disqualified" if the machine structurally collapses, falls over, or hangs or jams unpredictably (for more than 2 minutes) with no termination display, or damages a box, or terminates the operation before delivering at least one packed and closed box (with minimum 5 dowels in each, even if not placed or sorted correctly) to the pickup location/bin, or does not display termination/completion message on the LCD at the end of operation, or the team declares the termination. If any of the above happens to the first run, the team will have 3 minutes to fix the system and run for the second time, should they wish.

t. Each team will have a period of maximum 2 minutes to set up the machine before each run. If the preparation time exceeds 2 minutes, the run will be "disqualified."

u. There will be no control on the conditions of the contest environment.

v. The machine must pose no hazard to the operator, and shall not be perceived as hazardous (e.g., too much vibration or noise or frequent sparks during the operation is perceived as dangerous.)

Extra Design Features

The following features would enhance the machine performance, and increase the Bonus Points:

- **Robustness and Durability:** Machine is durably constructed and functions consistently with a small failure frequency and under different conditions of the operation environment.
- **Operability and Sustainability:** Little time/effort is needed to set up and calibrate the machine, and the machine is modular so that parts can be replaced or repaired easily.
- **Elegance and Safety:** Machine looks elegant, and operates quietly and smoothly with little or no sensible noise or vibration.
- **Extendibility:** Machine can accept dowels and boxes of different sizes and can package more than 3 boxes in each operation with little or no need for modifications.
- **Dexterity:** Machine can perform extra functions, such as labeling, box stacking, etc.
- **Compactness and Portability:** The entire prototype weighs not more than half of the maximum permitted weight and fits within a cubic envelope whose side is 75% of that of the maximum allowed envelope.
- **Real-time Date/Time Display:** Date and time of each inspection are displayed on the LCD in standby mode.
- **Permanent Logs:** Machine stores log information in permanent (EEPROM) memory.
- **PC Interface:** The operation information can be readily downloaded on a PC.

Expected Outcomes

Design and Construction Process: The team must follow a logical and systematic process in accomplishing their tasks of design, analysis, and construction. Conceptual design and system analysis are important steps of this project where the team has to compromise speed, accuracy, reliability, and cost. The detailed process must be reflected in the final report submitted by the team.

Proposal: Each team must work together to generate a proposal documentation on the design. The design proposal should reflect the conceptual design phase, team and project management with the scheduling, the steps to be taken for the detailed design and prototype fabrication, and the methods of manufacturing, integration and debugging to be followed in building the prototype.

Final Report: The final report details the entire process of detail-design, analysis, fabrication, and evaluation.

***Final Prototype*:** The final prototype developed by the team should reflect the work presented in the proposal. Any major or significant change in the design of the prototype after submitting the proposal must be agreed upon by the client and justified in the final report. The quality of the prototype may vary widely depending on the background of the team, the difficulty of the concept, and other limitations. Many of the deficiencies of these prototypes can be resolved later in the students' academic career.

Team Dynamics: The team must propose a solution and the plan in the proposal, and remain *loyal* to the proposal during the entire process. Hence, a close interaction between members of the team is required initially to be able to "*plan ahead.*" Early team dynamics may be strained, but interaction increases as the construction and integration of the machine proceed. Maximum team interaction occurs during the system integration, test and demonstration. The instructor will enhance the team dynamics by spending some time with the teams examining the progress. In many cases students remember this team experience (including their teammates) when they are seniors, or even when they are returning alumni. Professional and humane characteristics are expected in all team activities.

Grade evaluation will be heavily weighted to the generated design concepts, proposal, final report, and the way each individual/team has interacted and performed the tasks. Nevertheless, the final product and performance evaluation (competition) will remain as crucial portions in the overall grade.

Statement of Work

Each team is composed of three students. Conceptual design, system analysis, project planning, and system integration and debugging must be performed through a close interaction of all members of the team. However, for the sake of implementation, tasks can be broken into the following categories:

Processing and Control (Microcontroller)
One student shall program all the software for the system. In addition to combinatorial and sequential logic required for the operation algorithm, the keypad and display interface with the microcontroller is also part of this assignment. Some extra coding may also be needed for system debugging. Further utilization of the microcontroller may be needed if the team plans to accomplish some of the Extra Design Features, such as Real-time Date/Time Display, Permanent Logs, and PC Interface. For a low-power, high-end microcontroller, the assembly language would be the most efficient option for programming. Some cross-assemblers can translate C and/or Basic into machine code resulting in a less efficient and tractable code, to a degree that it may deteriorate system functionality. For the processing hardware, the use of the microcontroller board in the Project Kit is permitted if budget allows. Otherwise, the microcontroller student has the responsibility of assembling the microcontroller board. It is required that the microcontroller be functional for basic design features and programmable by the Reading Week, so that system integration and testing may begin right after the Reading Week. Often integration requires additional adjustments to the processor hardware and software. In addition, after the Reading week, the person responsible for the Microcontroller subsystem shall effectively assist the Electromechanical subsystem with duplication or fabrication of components and subassemblies. **The division of Electromechanical tasks among all members must be specified clearly in the project proposal.**

Mechanism and Actuation (Electromechanical)
One student shall be primarily responsible for constructing the structure and frames and incorporating whatever actuators and mechanisms are required in the machine. Nonetheless, after the Reading Week, the Microcontroller and Circuit members will join the Electromechanical member for completing the tasks, according to the plan specified in the proposal. Major subsystems of the Electromechanical category can include: frame and structure, dowel positioning and dispensing, and box closing mechanisms. In addition to design and analysis of these subsystems, their fabrication and/or assembly as well as assigning the locations of the sensors and boards are also parts of the Electromechanical category. Some off-the-shelf mechanisms or platforms can also be used for the above-mentioned subsystems, but this must be clearly addressed in the proposal and authorized by the instructor. Although integration of the entire system might seem as a "mechanical" task by nature, all members of the team should equally and effectively take part in the integration process.

Instrumentation and Interfacing (Circuit)
One student shall construct all the digital and analog interfacing electronics to connect the sensors and actuators to the microcontroller board. This includes motor/solenoid driver circuits. All sensors and input/output signal calibration/protection are also part of this category. In those situations where the primary calibration for a transducer is positional in nature, such as a stop switch, the task is still part of the Circuit subsystem, but consultation with the Electromechanical member is advised. For the actuator drivers, the use of driver board in the Project Kit or driver IC's is permitted if the budget allows, but the Circuit member must design and build at least one "open" circuit for a motor (DC or Stepper) in the system and prove their functionality. Dowel detection and position sensing (and possible shaft encoding) could be the major sensory tasks of this subsystem, in addition to the driver circuits and cabling. The Circuit member shall also acquire suitable power supplies for the actuators, circuits, sensors, and the microcontroller. Further, after the Reading Week, the person responsible for circuits shall effectively assist the Electromechanical subsystem with duplication or fabrication of components and subassemblies. **The division of Electromechanical tasks among all members must be specified clearly in the project proposal.**

Discussion

In this design, speed, accuracy, reliability, and cost are competing factors. Designers should first analyze the performance criteria to specify the level of acceptable compromise in each of the above factors. A variety of solutions can be proposed for transporting the dowels from the reservoir to the box, for counting the dowels, and for closing the covers. Hence, a careful analysis of the force and power required for each function is important.

Students might encounter problems with manufacturing the product. With limited experience in shop practices, final prototypes may not always work as anticipated. This can be frustrating to the students. As with any life experience, product fabrication will improve as the students gain maturity, not only in shop activities, but also in the engineering science background. The contest session provides a proof of the paper design. It also demonstrates to students that in real life the result does not always follow the prediction of theory. This is a good time to remind the students that ***"what we have to learn to do, we learn by doing."***

Appendix E - RELEVANT DATASHEETS

MICROCHIP **PIC18F2525/2620/4525/4620**

28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology

Power Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 2.5 µA typical
- Sleep mode current down to 100 nA typical
- Timer1 Oscillator: 1.8 µA, 32 kHz, 2V
- Watchdog Timer: 1.4 µA, 2V typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, up to 40 MHz
- 4x Phase Lock Loop (PLL) – available for crystal and internal oscillators)
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 MHz
- Internal oscillator block:
  - 8 user selectable frequencies, from 31 kHz to 8 MHz
  - Provides a complete range of clock speeds from 31 kHz to 32 MHz when used with PLL
  - User tunable to compensate for frequency drift
- Secondary oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor
  - Allows for safe shutdown if peripheral clock stops

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- Four input change interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
  - One, two or four PWM outputs
  - Selectable polarity
  - Programmable dead time
  - Auto-Shutdown and Auto-Restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI™ (all 4 modes) and I^2C™ Master and Slave modes
- Enhanced Addressable USART module:
  - Supports RS-485, RS-232 and LIN 1.2
  - RS-232 operation using internal oscillator block (no external crystal required)
  - Auto-Wake-up on Start bit
  - Auto-Baud Detect
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D):
  - Auto-acquisition capability
  - Conversion available during Sleep
- Dual analog comparators with input multiplexing
- Programmable 16-level High/Low-Voltage Detection (HLVD) module:
  - Supports interrupt on High/Low-Voltage Detection

Special Microcontroller Features:

- C compiler optimized architecture:
  - Optional extended instruction set designed to optimize re-entrant code
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Flash/Data EEPROM Retention: 100 years typical
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 4 ms to 131s
- Single-supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Wide operating voltage range: 2.0V to 5.5V
- Programmable Brown-out Reset (BOR) with software enable option

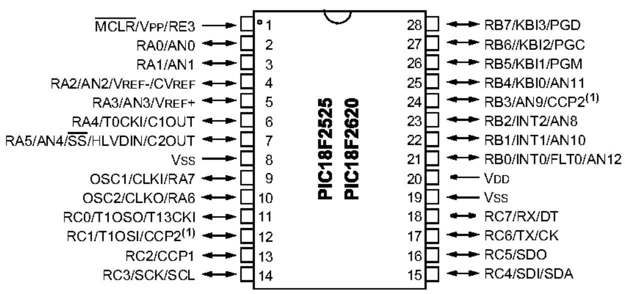
Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI™	Master I^2C™			
PIC18F2525	48K	24576	3986	1024	25	10	2/0	Y	Y	1	2	1/3
PIC18F2620	64K	32768	3986	1024	25	10	2/0	Y	Y	1	2	1/3
PIC18F4525	48K	24576	3986	1024	36	13	1/1	Y	Y	1	2	1/3
PIC18F4620	64K	32768	3986	1024	36	13	1/1	Y	Y	1	2	1/3

 Preliminary

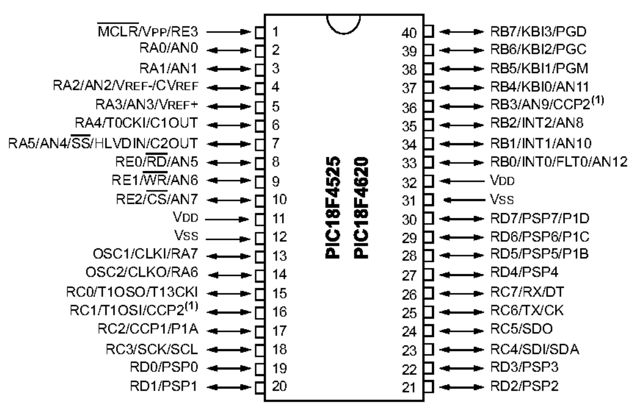
PIC18F2525/2620/4525/4620

Pin Diagrams

28-Pin SPDIP, SOIC



40-Pin PDIP



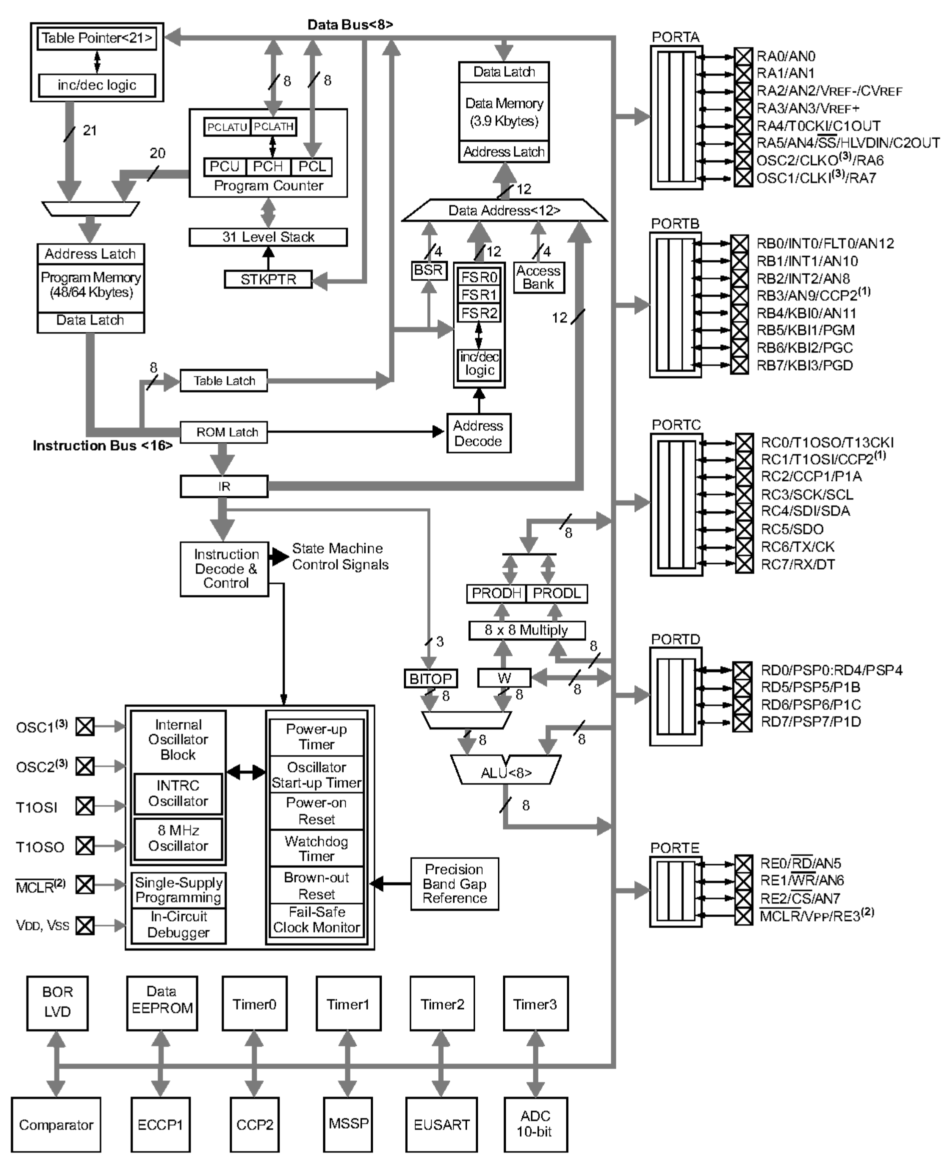
Note 1: RB3 is the alternate pin for CCP2 multiplexing.

PIC18F2525/2620/4525/4620

TABLE 1-1: DEVICE FEATURES

Features	PIC18F2525	PIC18F2620	PIC18F4525	PIC18F4620
Operating Frequency	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz
Program Memory (Bytes)	49152	65536	49152	65536
Program Memory (Instructions)	24576	32768	24576	32768
Data Memory (Bytes)	3968	3968	3968	3968
Data EEPROM Memory (Bytes)	1024	1024	1024	1024
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/Compare/ PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Parallel Communications (PSP)	No	No	Yes	Yes
10-bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin SPDIP 28-pin SOIC	28-pin SPDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

FIGURE 1-2: PIC18F4525/4620 (40/44-PIN) BLOCK DIAGRAM



Note 1: CCP2 is multiplexed with RC1 when configuration bit CCP2MX is set, or RB3 when CCP2MX is not set.

2: RE3 is only available when $\overline{MCLR}$ functionality is disabled.

3: OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to **Section 2.0 "Oscillator Configurations"** for additional information.

PIC18F2525/2620/4525/4620

TABLE 1-3: PIC18F4525/4620 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTA is a bidirectional I/O port.
RA0/AN0	2	19	19			
RA0				I/O	TTL	Digital I/O.
AN0				I	Analog	Analog input 0.
RA1/AN1	3	20	20			
RA1				I/O	TTL	Digital I/O.
AN1				I	Analog	Analog input 1.
RA2/AN2/VREF-/CVREF	4	21	21			
RA2				I/O	TTL	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (low) input.
CVREF				O	Analog	Comparator reference voltage output.
RA3/AN3/VREF+	5	22	22			
RA3				I/O	TTL	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (high) input.
RA4/T0CKI/C1OUT	6	23	23			
RA4				I/O	ST	Digital I/O.
T0CKI				I	ST	Timer0 external clock input.
C1OUT				O	—	Comparator 1 output.
RA5/AN4/$\overline{SS}$/HLVDIN/C2OUT	7	24	24			
RA5				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
$\overline{SS}$				I	TTL	SPI™ slave select input.
HLVDIN				I	Analog	High/Low-Voltage Detect input.
C2OUT				O	—	Comparator 2 output.
RA6						See the OSC2/CLKO/RA6 pin.
RA7						See the OSC1/CLKI/RA7 pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Default assignment for CCP2 when configuration bit CCP2MX is set.

2: Alternate assignment for CCP2 when configuration bit CCP2MX is cleared.

PIC18F2525/2620/4525/4620

TABLE 1-3: PIC18F4525/4620 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/INT0/FLT0/AN12	33	9	8			
RB0				I/O	TTL	Digital I/O.
INT0				I	ST	External interrupt 0.
FLT0				I	ST	PWM Fault input for Enhanced CCP1.
AN12				I	Analog	Analog input 12.
RB1/INT1/AN10	34	10	9			
RB1				I/O	TTL	Digital I/O.
INT1				I	ST	External interrupt 1.
AN10				I	Analog	Analog input 10.
RB2/INT2/AN8	35	11	10			
RB2				I/O	TTL	Digital I/O.
INT2				I	ST	External interrupt 2.
AN8				I	Analog	Analog input 8.
RB3/AN9/CCP2	36	12	11			
RB3				I/O	TTL	Digital I/O.
AN9				I	Analog	Analog input 9.
CCP2[1]				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
RB4/KBI0/AN11	37	14	14			
RB4				I/O	TTL	Digital I/O.
KBI0				I	TTL	Interrupt-on-change pin.
AN11				I	Analog	Analog input 11.
RB5/KBI1/PGM	38	15	15			
RB5				I/O	TTL	Digital I/O.
KBI1				I	TTL	Interrupt-on-change pin.
PGM				I/O	ST	Low-Voltage ICSP™ Programming enable pin.
RB6/KBI2/PGC	39	16	16			
RB6				I/O	TTL	Digital I/O.
KBI2				I	TTL	Interrupt-on-change pin.
PGC				I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD	40	17	17			
RB7				I/O	TTL	Digital I/O.
KBI3				I	TTL	Interrupt-on-change pin.
PGD				I/O	ST	In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; O = Output; CMOS = CMOS compatible input or output; I = Input; P = Power

Note 1: Default assignment for CCP2 when configuration bit CCP2MX is set.
2: Alternate assignment for CCP2 when configuration bit CCP2MX is cleared.

TABLE 1-3: PIC18F4525/4620 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number PDIP	QFN	TQFP	Pin Type	Buffer Type	Description
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	15	34	32			
RC0				I/O	ST	Digital I/O.
T1OSO				O	—	Timer1 oscillator output.
T13CKI				I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2	16	35	35			
RC1				I/O	ST	Digital I/O.
T1OSI				I	CMOS	Timer1 oscillator input.
CCP2(2)				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
RC2/CCP1/P1A	17	36	36			
RC2				I/O	ST	Digital I/O.
CCP1				I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output.
P1A				O	—	Enhanced CCP1 output.
RC3/SCK/SCL	18	37	37			
RC3				I/O	ST	Digital I/O.
SCK				I/O	ST	Synchronous serial clock input/output for SPI™ mode.
SCL				I/O	ST	Synchronous serial clock input/output for I^2C™ mode.
RC4/SDI/SDA	23	42	42			
RC4				I/O	ST	Digital I/O.
SDI				I	ST	SPI data in.
SDA				I/O	ST	I^2C data I/O.
RC5/SDO	24	43	43			
RC5				I/O	ST	Digital I/O.
SDO				O	—	SPI data out.
RC6/TX/CK	25	44	44			
RC6				I/O	ST	Digital I/O.
TX				O	—	EUSART asynchronous transmit.
CK				I/O	ST	EUSART synchronous clock (see related RX/DT).
RC7/RX/DT	26	1	1			
RC7				I/O	ST	Digital I/O.
RX				I	ST	EUSART asynchronous receive.
DT				I/O	ST	EUSART synchronous data (see related TX/CK).

Legend: TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; O = Output; CMOS = CMOS compatible input or output; I = Input; P = Power

Note 1: Default assignment for CCP2 when configuration bit CCP2MX is set.
2: Alternate assignment for CCP2 when configuration bit CCP2MX is cleared.

PIC18F2525/2620/4525/4620

TABLE 1-3: PIC18F4525/4620 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTD is a bidirectional I/O port or a Parallel Slave Port (PSP) for interfacing to a microprocessor port. These pins have TTL input buffers when the PSP module is enabled.
RD0/PSP0	19	38	38			
RD0				I/O	ST	Digital I/O.
PSP0				I/O	TTL	Parallel Slave Port data.
RD1/PSP1	20	39	39			
RD1				I/O	ST	Digital I/O.
PSP1				I/O	TTL	Parallel Slave Port data.
RD2/PSP2	21	40	40			
RD2				I/O	ST	Digital I/O.
PSP2				I/O	TTL	Parallel Slave Port data.
RD3/PSP3	22	41	41			
RD3				I/O	ST	Digital I/O.
PSP3				I/O	TTL	Parallel Slave Port data.
RD4/PSP4	27	2	2			
RD4				I/O	ST	Digital I/O.
PSP4				I/O	TTL	Parallel Slave Port data.
RD5/PSP5/P1B	28	3	3			
RD5				I/O	ST	Digital I/O.
PSP5				I/O	TTL	Parallel Slave Port data.
P1B				O	—	Enhanced CCP1 output.
RD6/PSP6/P1C	29	4	4			
RD6				I/O	ST	Digital I/O.
PSP6				I/O	TTL	Parallel Slave Port data.
P1C				O	—	Enhanced CCP1 output.
RD7/PSP7/P1D	30	5	5			
RD7				I/O	ST	Digital I/O.
PSP7				I/O	TTL	Parallel Slave Port data.
P1D				O	—	Enhanced CCP1 output.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels; I = Input
O = Output; P = Power

Note 1: Default assignment for CCP2 when configuration bit CCP2MX is set.
2: Alternate assignment for CCP2 when configuration bit CCP2MX is cleared.

 Preliminary

TABLE 1-3: PIC18F4525/4620 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTE is a bidirectional I/O port.
RE0/$\overline{\text{RD}}$/AN5	8	25	25			
RE0				I/O	ST	Digital I/O.
$\overline{\text{RD}}$				I	TTL	Read control for Parallel Slave Port (see also $\overline{\text{WR}}$ and $\overline{\text{CS}}$ pins).
AN5				I	Analog	Analog input 5.
RE1/$\overline{\text{WR}}$/AN6	9	26	26			
RE1				I/O	ST	Digital I/O.
$\overline{\text{WR}}$				I	TTL	Write control for Parallel Slave Port (see $\overline{\text{CS}}$ and $\overline{\text{RD}}$ pins).
AN6				I	Analog	Analog input 6.
RE2/$\overline{\text{CS}}$/AN7	10	27	27			
RE2				I/O	ST	Digital I/O.
$\overline{\text{CS}}$				I	TTL	Chip Select control for Parallel Slave Port (see related $\overline{\text{RD}}$ and $\overline{\text{WR}}$).
AN7				I	Analog	Analog input 7.
RE3	—	—	—	—	—	See $\overline{\text{MCLR}}$/V_{PP}/RE3 pin.
V_{SS}	12, 31	6, 30, 31	6, 29	P	—	Ground reference for logic and I/O pins.
V_{DD}	11, 32	7, 8, 28, 29	7, 28	P	—	Positive supply for logic and I/O pins.
NC	—	13	12, 13, 33, 34	—	—	No connect.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Default assignment for CCP2 when configuration bit CCP2MX is set.

2: Alternate assignment for CCP2 when configuration bit CCP2MX is cleared.

4.2 Master Clear ($\overline{\text{MCLR}}$)

The $\overline{\text{MCLR}}$ pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the $\overline{\text{MCLR}}$ Reset path which detects and ignores small pulses.

The $\overline{\text{MCLR}}$ pin is not driven low by any internal Resets, including the WDT.

In PIC18F2525/2620/4525/4620 devices, the $\overline{\text{MCLR}}$ input can be disabled with the MCLRE configuration bit. When $\overline{\text{MCLR}}$ is disabled, the pin becomes a digital input. See **Section 10.5 "PORTE, TRISE and LATE Registers"** for more information.

4.3 Power-on Reset (POR)

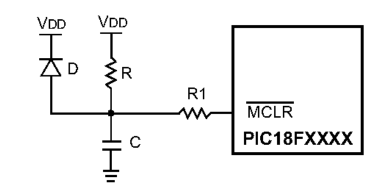
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the $\overline{\text{MCLR}}$ pin through a resistor (1 kΩ to 10 kΩ) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004). For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the $\overline{\text{POR}}$ bit (RCON<1>). The state of the bit is set to '0' whenever a POR occurs; it does not change for any other Reset event. $\overline{\text{POR}}$ is not reset to '1' by any hardware event. To capture multiple events, the user manually resets the bit to '1' in software following any POR.

FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

3: R1 ≥ 1 kΩ will limit any current flowing into $\overline{\text{MCLR}}$ from external capacitor C, in the event of $\overline{\text{MCLR}}$/VPP pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

5.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 Enhanced microcontroller devices:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in **Section 6.0 "Flash Program Memory"**. Data EEPROM is discussed separately in **Section 7.0 "Data EEPROM Memory"**.

5.1 Program Memory Organization

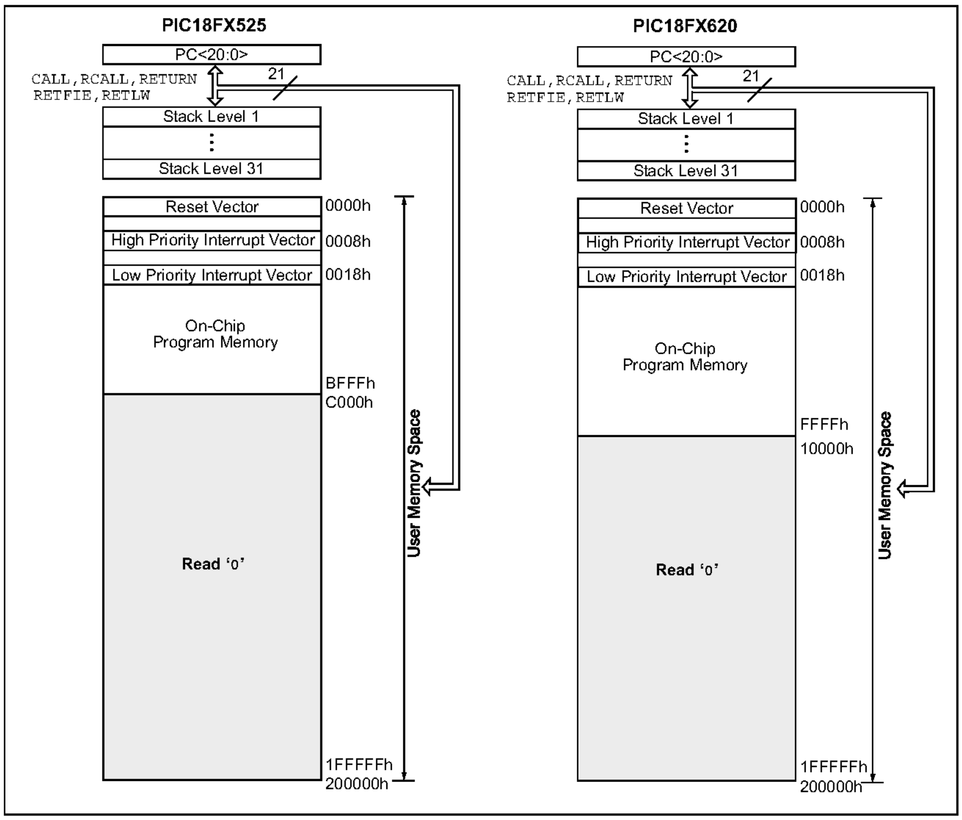
PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

The PIC18F2525 and PIC18F4525 each have 48 Kbytes of Flash memory and can store up to 24,576 single-word instructions. The PIC18F2620 and PIC18F4620 each have 64 Kbytes of Flash memory and can store up to 32,768 single-word instructions.

PIC18 devices have two interrupt vectors. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

The program memory maps for PIC18FX525 and PIC18FX620 devices are shown in Figure 5-1.

FIGURE 5-1: PROGRAM MEMORY MAP AND STACK FOR PIC18F2525/2620/4525/4620 DEVICES



5.2 PIC18 Instruction Cycle

5.2.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by four to generate four non-overlapping quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the program counter is incremented on every Q1; the instruction is fetched from the program memory and latched into the instruction register during Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 5-3.

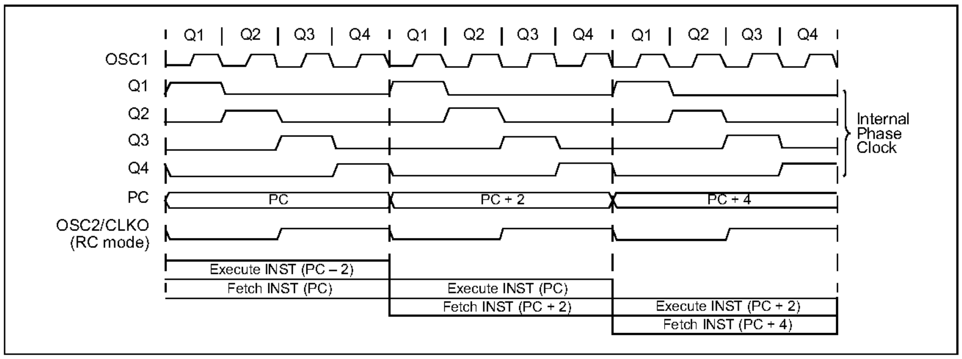
5.2.2 INSTRUCTION FLOW/PIPELINING

An "Instruction Cycle" consists of four Q cycles: Q1 through Q4. The instruction fetch and execute are pipelined in such a manner that a fetch takes one instruction cycle, while the decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., `GOTO`), then two cycles are required to complete the instruction (Example 5-3).

A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 5-3: CLOCK/INSTRUCTION CYCLE



EXAMPLE 5-3: INSTRUCTION PIPELINE FLOW

	TCY0	TCY1	TCY2	TCY3	TCY4	TCY5
`1. MOVLW 55h`	Fetch 1	Execute 1				
`2. MOVWF PORTB`		Fetch 2	Execute 2			
`3. BRA SUB_1`			Fetch 3	Execute 3		
`4. BSF PORTA, BIT3 (Forced NOP)`				Fetch 4	Flush (`NOP`)	
`5. Instruction @ address SUB_1`					Fetch SUB_1	Execute SUB_1

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

7.0 DATA EEPROM MEMORY

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, that is used for long-term storage of program data. It is not directly mapped in either the register file or program memory space but is indirectly addressed through the Special Function Registers (SFRs). The EEPROM is readable and writable during normal operation over the entire V_{DD} range.

Five SFRs are used to read and write to the data EEPROM as well as the program memory. They are:

- EECON1
- EECON2
- EEDATA
- EEADR
- EEADRH

The data EEPROM allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and the EEADRH:EEADR register pair holds the address of the EEPROM location being accessed.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer; it will vary with voltage and temperature as well as from chip to chip. Please refer to parameter D122 (Table 26-1 in **Section 26.0 "Electrical Characteristics"**) for exact limits.

7.1 EEADR and EEADRH Registers

The EEADRH:EEADR register pair is used to address the data EEPROM for read and write operations. EEADRH holds the two MSbits of the address; the upper 6 bits are ignored. The 10-bit range of the pair can address a memory range of 1024 bytes (00h to 3FFh).

7.2 EECON1 and EECON2 Registers

Access to the data EEPROM is controlled by two registers: EECON1 and EECON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The EECON1 register (Register 7-1) is the control register for data and program memory access. Control bit EEPGD determines if the access will be to program or data EEPROM memory. When clear, operations will access the data EEPROM memory. When set, program memory is accessed.

Control bit CFGS determines if the access will be to the configuration registers or to program memory/data EEPROM memory. When set, subsequent operations access configuration registers. When CFGS is clear, the EEPGD bit selects either program Flash or data EEPROM memory.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WREN bit is set and cleared when the internal programming timer expires and the write operation is complete.

Note:	During normal operation, the WRERR is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset, or a write operation was attempted improperly.

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software; it is cleared in hardware at the completion of the write operation.

Note:	The EEIF interrupt flag bit (PIR2<4>) is set when the write is complete. It must be cleared in software.

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (EEPGD = 1). Program memory is read using table read instructions. See **Section 6.1 "Table Reads and Table Writes"** regarding table reads.

The EECON2 register is not a physical register. It is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

9.0 INTERRUPTS

The PIC18F2525/2620/4525/4620 devices have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high priority level or a low priority level. The high priority interrupt vector is at 0008h and the low priority interrupt vector is at 0018h. High priority interrupt events will interrupt any low priority interrupts that may be in progress.

There are ten registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- **Flag bit** to indicate that an interrupt event occurred
- **Enable bit** that allows program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 0008h or 0018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PICmicro® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit, which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit, which enables/disables all interrupt sources. All interrupts branch to address 0008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High priority interrupt sources can interrupt a low priority interrupt. Low priority interrupts are not processed while high priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The “return from interrupt” instruction, `RETFIE`, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used), which re-enables interrupts.

For external interrupt events, such as the INT pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding enable bit or the GIE bit.

Note:	Do not use the `MOVFF` instruction to modify any of the interrupt control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

15.0 CAPTURE/COMPARE/PWM (CCP) MODULES

PIC18F2525/2620/4525/4620 devices all have two CCP (Capture/Compare/PWM) modules. Each module contains a 16-bit register which can operate as a 16-bit Capture register, a 16-bit Compare register or a PWM Master/Slave Duty Cycle register.

In 28-pin devices, the two standard CCP modules (CCP1 and CCP2) operate as described in this chapter. In 40/44-pin devices, CCP1 is implemented as an Enhanced CCP module with standard Capture and Compare modes and Enhanced PWM modes. The ECCP implementation is discussed in **Section 16.0 "Enhanced Capture/Compare/PWM (ECCP) Module"**.

The Capture and Compare operations described in this chapter apply to all standard and Enhanced CCP modules.

Note:	Throughout this section and **Section 16.0 "Enhanced Capture/Compare/PWM (ECCP) Module"**, references to the register and bit names for CCP modules are referred to generically by the use of 'x' or 'y' in place of the specific module number. Thus, "CCPxCON" might refer to the control register for CCP1, CCP2 or ECCP1. "CCPxCON" is used throughout these sections to refer to the module control register, regardless of whether the CCP module is a standard or enhanced implementation.

REGISTER 15-1: CCPxCON REGISTER (CCP2 MODULE, CCP1 MODULE IN 28-PIN DEVICES)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0

bit 7 bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DCxB1:DCxB0**: PWM Duty Cycle bit 1 and bit 0 for CCP Module x

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the two LSbs (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight MSbs (DCx9:DCx2) of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0**: CCP Module x Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCP module)
0001 = Reserved
0010 = Compare mode, toggle output on match (CCPIF bit is set)
0011 = Reserved
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode: initialize CCP pin low; on compare match, force CCP pin high (CCPxIF bit is set)
1001 = Compare mode: initialize CCP pin high; on compare match, force CCP pin low (CCPxIF bit is set)
1010 = Compare mode: generate software interrupt on compare match (CCPxIF bit is set, CCP pin reflects I/O state)
1011 = Compare mode: trigger special event, reset timer, start A/D conversion on CCP2 match (CCPxIF bit is set)
11xx = PWM mode

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

PIC18F2525/2620/4525/4620

15.4 PWM Mode

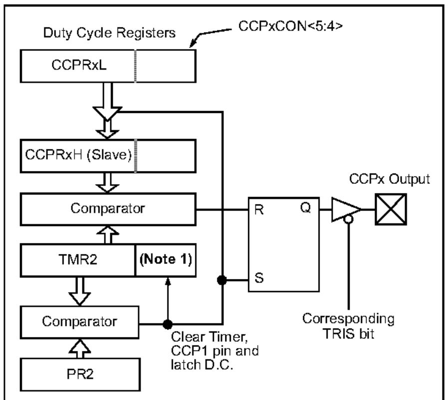
In Pulse-Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP2 pin is multiplexed with a PORTB or PORTC data latch, the appropriate TRIS bit must be cleared to make the CCP2 pin an output.

> **Note:** Clearing the CCP2CON register will force the RB3 or RC1 output latch (depending on device configuration) to the default low level. This is not the PORTB or PORTC I/O data latch.

Figure 15-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.4.4 "Setup for PWM Operation"**.

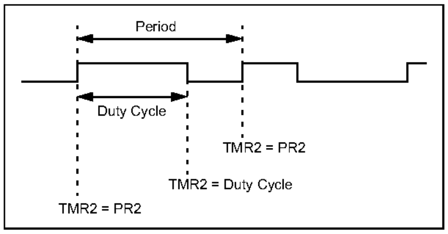
FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

A PWM output (Figure 15-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 15-4: PWM OUTPUT



15.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

EQUATION 15-1:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
- The PWM duty cycle is latched from CCPRxL into CCPRxH

> **Note:** The Timer2 postscalers (see **Section 13.0 "Timer2 Module"**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

15.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPRxL register and to the CCPxCON<5:4> bits. Up to 10-bit resolution is available. The CCPRxL contains the eight MSbs and the CCPxCON<5:4> contains the two LSbs. This 10-bit value is represented by CCPRxL:CCPxCON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

EQUATION 15-2:

$$\text{PWM Duty Cycle} = (\text{CCPRxL:CCPxCON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

CCPRxL and CCPxCON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR2H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPRxH is a read-only register.

The CCPR2H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPRxH and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP2 pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

EQUATION 15-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCP2 pin will not be cleared.

TABLE 15-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

15.4.3 PWM AUTO-SHUTDOWN (CCP1 ONLY)

The PWM auto-shutdown features of the Enhanced CCP module are also available to CCP1 in 28-pin devices. The operation of this feature is discussed in detail in **Section 16.4.7 "Enhanced PWM Auto-Shutdown"**.

Auto-shutdown features are not available for CCP2.

15.4.4 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
5. Configure the CCPx module for PWM operation.

PIC18F2525/2620/4525/4620

TABLE 15-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
RCON	IPEN	SBOREN(1)	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	48
PIR1	PSPIF(2)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE(2)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP(2)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TRISB	PORTB Data Direction Control Register								52
TRISC	PORTC Data Direction Control Register								52
TMR2	Timer2 Register								50
PR2	Timer2 Period Register								50
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	50
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								51
CCPR1H	Capture/Compare/PWM Register 1 High Byte								51
CCP1CON	P1M1(2)	P1M0(2)	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	51
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								51
CCPR2H	Capture/Compare/PWM Register 2 High Byte								51
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	51
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1(2)	PSSBD0(2)	51
PWM1CON	PRSEN	PDC6(2)	PDC5(2)	PDC4(2)	PDC3(2)	PDC2(2)	PDC1(2)	PDC0(2)	51

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PWM or Timer2.

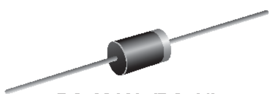
Note 1: The SBOREN bit is only available when the BOREN1:BOREN0 configuration bits = 01; otherwise, it is disabled and reads as '0'. See **Section 4.4 "Brown-out Reset (BOR)"**.

2: These bits are unimplemented on 28-pin devices and read as '0'.

1N4001 thru 1N4007

Vishay General Semiconductor

General Purpose Plastic Rectifier



DO-204AL (DO-41)

FEATURES

- Low forward voltage drop
- Low leakage current
- High forward surge capability
- Solder dip 275 °C max. 10 s, per JESD 22-B106
- Compliant to RoHS Directive 2002/95/EC and in accordance to WEEE 2002/96/EC

RoHS COMPLIANT

TYPICAL APPLICATIONS

For use in general purpose rectification of power supplies, inverters, converters and freewheeling diodes application.

Note

- These devices are not AEC-Q101 qualified.

MECHANICAL DATA

Case: DO-204AL, molded epoxy body
Molding compound meets UL 94 V-0 flammability rating
Base P/N-E3 - RoHS compliant, commercial grade

Terminals: Matte tin plated leads, solderable per J-STD-002 and JESD 22-B102
E3 suffix meets JESD 201 class 1A whisker test

Polarity: Color band denotes cathode end

PRIMARY CHARACTERISTICS	
$I_{F(AV)}$	1.0 A
V_{RRM}	50 V to 1000 V
I_{FSM} (8.3 ms sine-wave)	30 A
I_{FSM} (square wave t_p = 1 ms)	45 A
V_F	1.1 V
I_R	5.0 µA
T_J max.	150 °C

MAXIMUM RATINGS (T_A = 25 °C unless otherwise noted)

PARAMETER		SYMBOL	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	UNIT
Maximum repetitive peak reverse voltage		V_{RRM}	50	100	200	400	600	800	1000	V
Maximum RMS voltage		V_{RMS}	35	70	140	280	420	560	700	V
Maximum DC blocking voltage		V_{DC}	50	100	200	400	600	800	1000	V
Maximum average forward rectified current 0.375" (9.5 mm) lead length at T_A = 75 °C		$I_{F(AV)}$	1.0							A
Peak forward surge current 8.3 ms single half sine-wave superimposed on rated load		I_{FSM}	30							A
Non-repetitive peak forward surge current square waveform T_A = 25 °C (fig. 3)	t_p = 1 ms	I_{FSM}	45							A
	t_p = 2 ms		35							
	t_p = 5 ms		30							
Maximum full load reverse current, full cycle average 0.375" (9.5 mm) lead length T_L = 75 °C		$I_{R(AV)}$	30							µA
Rating for fusing (t < 8.3 ms)		I^2t [1]	3.7							A^2s
Operating junction and storage temperature range		T_J, T_{STG}	- 50 to + 150							°C

Note

[1] For device using on bridge rectifier appliaction

1N4001 thru 1N4007

Vishay General Semiconductor

ELECTRICAL CHARACTERISTICS (T_A = 25 °C unless otherwise noted)											
PARAMETER	TEST CONDITIONS	SYMBOL	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	UNIT
Maximum instantaneous forward voltage	1.0 A	V_F	1.1							V
Maximum DC reverse current at rated DC blocking voltage	T_A = 25 °C	I_R	5.0							μA
	T_A = 125 °C		50							
Typical junction capacitance	4.0 V, 1 MHz	C_J	15							pF

THERMAL CHARACTERISTICS (T_A = 25 °C unless otherwise noted)									
PARAMETER	SYMBOL	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	UNIT
Typical thermal resistance	$R_{\theta JA}$ (1)	50							°C/W
	$R_{\theta JL}$ (1)	25							

Note

(1) Thermal resistance from junction to ambient at 0.375" (9.5 mm) lead length, PCB mounted

ORDERING INFORMATION (Example)				
PREFERRED P/N	UNIT WEIGHT (g)	PREFERRED PACKAGE CODE	BASE QUANTITY	DELIVERY MODE
1N4004-E3/54	0.33	54	5500	13" diameter paper tape and reel
1N4004-E3/73	0.33	73	3000	Ammo pack packaging

RATINGS AND CHARACTERISTICS CURVES

(T_A = 25 °C unless otherwise noted)

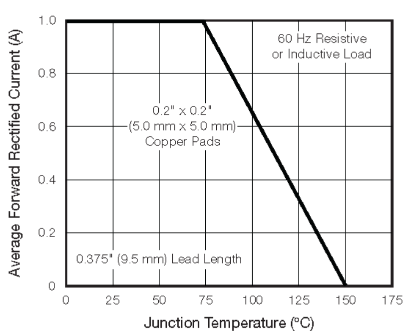


Fig. 1 - Forward Current Derating Curve

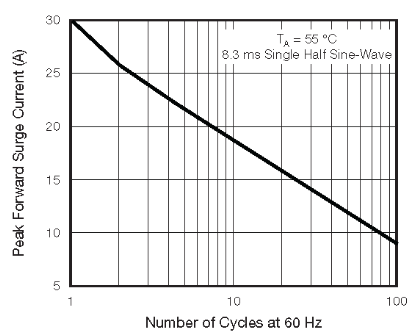


Fig. 2 - Maximum Non-repetitive Peak Forward Surge Current

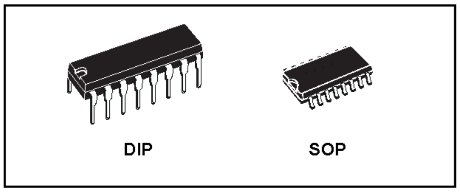
For technical questions within your region, please contact one of the following:
DiodesAmericas@vishay.com, DiodesAsia@vishay.com, DiodesEurope@vishay.com

Document Number: 88503
Revision: 23-Feb-11

HCF4050B

HEX BUFFER/CONVERTER (NON INVERTING)

- PROPAGATION DELAY TIME : t_{PD} = 40ns (TYP.) at V_{DD} = 10V C_L = 50pF
- HIGH TO LOW LEVEL LOGIC CONVERSION
- HIGH "SINK" AND "SOURCE" CURRENT CAPABILITY
- QUIESCENT CURRENT SPECIFIED UP TO 20V
- 5V, 10V AND 15V PARAMETRIC RATINGS
- INPUT LEAKAGE CURRENT I_I = 100nA (MAX) AT V_{DD} = 18V T_A = 25°C
- 100% TESTED FOR QUIESCENT CURRENT
- MEETS ALL REQUIREMENTS OF JEDEC JESD13B " STANDARD SPECIFICATIONS FOR DESCRIPTION OF B SERIES CMOS DEVICES"



ORDER CODES

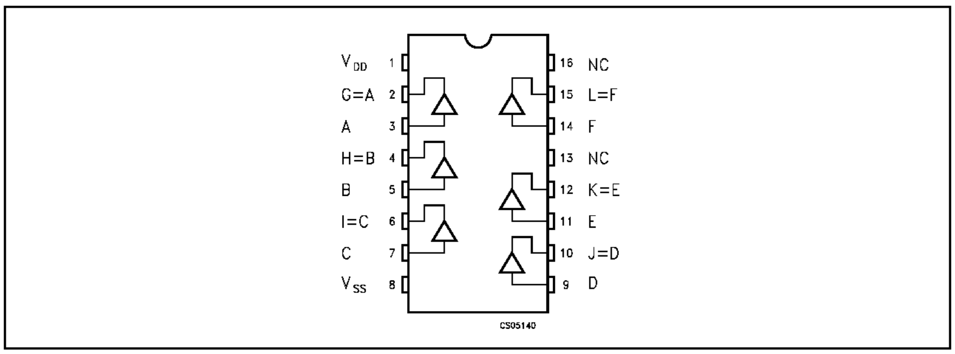
PACKAGE	TUBE	T & R
DIP	HCF4050BEY	
SOP	HCF4050BM1	HCF4050M013TR

DESCRIPTION

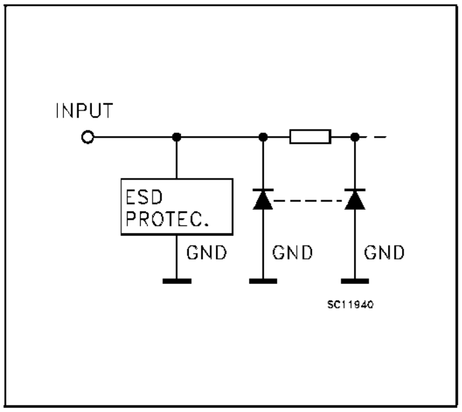
The HCF4050B is a monolithic integrated circuit fabricated in Metal Oxide Semiconductor technology available in DIP and SOP packages.
It is an non inverting Hex Buffer/Converter and feature logic level conversions using only one supply voltage (V_{DD}).

The input high level signal (V_{IH}) can exceed the V_{DD} supply voltage when these devices are used for logic level conversions. This device is intended for use as CMOS to DTL/TTL converters and can drive directly two DTL/TTL loads (V_{DD}=5V, V_{OL}≤0.4V and I_{OL}≤3.2mA.

PIN CONNECTION



INPUT EQUIVALENT CIRCUIT



PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
3, 5, 7, 9, 11, 14	A, B, C, D, E, F	Data Inputs
2, 4, 6, 10, 12, 15	G, H, I, J, K, L	Data Outputs
13 , 16	NC	Not Connected
8	V_{SS}	Negative Supply Voltage
1	V_{DD}	Positive Supply Voltage

TRUTH TABLE

INPUTS	OUTPUTS
A, B, C, D, E, F	**G, H, I, J, K, L**
L	L
H	H

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.5 to +22	V
V_I	DC Input Voltage	-0.5 to +18	V
I_I	DC Input Current	± 10	mA
P_D	Power Dissipation per Package	200	mW
	Power Dissipation per Output Transistor	100	mW
T_{op}	Operating Temperature	-55 to +125	°C
T_{stg}	Storage Temperature	-65 to +150	°C

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these conditions is not implied.
All voltage values are referred to V_{SS} pin voltage.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	3 to 20	V
V_I	Input Voltage	-0.5 to 15V	V
T_{op}	Operating Temperature	-55 to 125	°C

DC SPECIFICATIONS

Symbol	Parameter	Test Condition				Value							Unit
		V_I (V)	V_O (V)	$\|I_O\|$ (µA)	V_{DD} (V)	T_A = 25°C			-40 to 85°C		-55 to 125°C		
						Min.	Typ.	Max.	Min.	Max.	Min.	Max.	
I_L	Quiescent Current	0/5			5		0.02	1		30		30	µA
		0/10			10		0.02	2		60		60	
		0/15			15		0.02	4		120		120	
		0/20			20		0.04	20		600		600	
V_{OH}	High Level Output Voltage	0/5		<1	5	4.95			4.95		4.95		V
		0/10		<1	10	9.95			9.95		9.95		
		0/15		<1	15	14.95			14.95		14.95		
V_{OL}	Low Level Output Voltage	5/0		<1	5		0.05			0.05		0.05	V
		10/0		<1	10		0.05			0.05		0.05	
		15/0		<1	15		0.05			0.05		0.05	
V_{IH}	High Level Input Voltage		0.5/4.5	<1	5	3.5			3.5		3.5		V
			1/9	<1	10	7			7		7		
			1.5/13.5	<1	15	11			11		11		
V_{IL}	Low Level Input Voltage		4.5/0.5	<1	5			1.5		1.5		1.5	V
			9/1	<1	10			3		3		3	
			13.5/1.5	<1	15			4		4		4	
I_{OH}	Output Drive Current	0/5	2.5	<1	5	-1.25	-6.4		-0.42		-0.42		mA
		0/5	4.6	<1	5	-0.51	-1.6		-0.38		-0.38		
		0/10	9.5	<1	10	-1.25	-3.6		-1		-1		
		0/15	13.5	<1	15	-3.75	-12		-3		-3		
I_{OL}	Output Sink Current	0/5	0.4	<1	5	3.2	6.4		2.6		2.6		mA
		0/10	0.5	<1	10	8	16		6.6		6.6		
		0/15	1.5	<1	15	24	48		19		19		
I_I	Input Leakage Current	0/18	Any Input		18		$\pm 10^{-5}$	± 0.1		± 1		± 1	µA
C_I	Input Capacitance		Any Input				5	7.5					pF

The Noise Margin for both "1" and "0" level is: 1V min. with V_{DD}=5V, 2V min. with V_{DD}=10V, 2.5V min. with V_{DD}=15V

8. Limiting values

Table 4. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134). Voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions		Min	Max	Unit
V_{CC}	supply voltage			–0.5	+7	V
I_{IK}	input clamping current	$V_I < -0.5$ V or $V_I > V_{CC} + 0.5$ V	[1]	-	±20	mA
I_{OK}	output clamping current	$V_O < -0.5$ V or $V_O > V_{CC} + 0.5$ V	[1]	-	±20	mA
I_O	output current	-0.5 V $< V_O < V_{CC} + 0.5$ V		-	±25	mA
I_{CC}	supply current			-	50	mA
I_{GND}	ground current			–50	-	mA
T_{stg}	storage temperature			–65	+150	°C
P_{tot}	total power dissipation		[2]			
	DIP14 package			-	750	mW
	SO14, (T)SSOP14 and DHVQFN14 packages			-	500	mW

[1] The input and output voltage ratings may be exceeded if the input and output current ratings are observed.

[2] For DIP14 package: P_{tot} derates linearly with 12 mW/K above 70 °C.
For SO14 package: P_{tot} derates linearly with 8 mW/K above 70 °C.
For (T)SSOP14 packages: P_{tot} derates linearly with 5.5 mW/K above 60 °C.
For DHVQFN14 packages: P_{tot} derates linearly with 4.5 mW/K above 60 °C.

9. Recommended operating conditions

Table 5. Recommended operating conditions

Voltages are referenced to GND (ground = 0 V)

Symbol	Parameter	Conditions	74HC14			74HCT14			Unit
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	supply voltage		2.0	5.0	6.0	4.5	5.0	5.5	V
V_I	input voltage		0	-	V_{CC}	0	-	V_{CC}	V
V_O	output voltage		0	-	V_{CC}	0	-	V_{CC}	V
T_{amb}	ambient temperature		–40	+25	+125	–40	+25	+125	°C

10. Static characteristics

Table 6. Static characteristics

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	T_{amb} = 25 °C			T_{amb} = −40 °C to +85 °C		T_{amb} = −40 °C to +125 °C		Unit
			Min	Typ	Max	Min	Max	Min	Max	
74HC14										
V_{OH}	HIGH-level output voltage	$V_I = V_{T+}$ or V_{T-}								
		$I_O = -20$ µA; V_{CC} = 2.0 V	1.9	2.0	-	1.9	-	1.9	-	V
		$I_O = -20$ µA; V_{CC} = 4.5 V	4.4	4.5	-	4.4	-	4.4	-	V
		$I_O = -20$ µA; V_{CC} = 6.0 V	5.9	6.0	-	5.9	-	5.9	-	V
		$I_O = -4.0$ mA; V_{CC} = 4.5 V	3.98	4.32	-	3.84	-	3.7	-	V
		$I_O = -5.2$ mA; V_{CC} = 6.0 V	5.48	5.81	-	5.34	-	5.2	-	V
V_{OL}	LOW-level output voltage	$V_I = V_{T+}$ or V_{T-}								
		I_O = 20 µA; V_{CC} = 2.0 V	-	0	0.1	-	0.1	-	0.1	V
		I_O = 20 µA; V_{CC} = 4.5 V	-	0	0.1	-	0.1	-	0.1	V
		I_O = 20 µA; V_{CC} = 6.0 V	-	0	0.1	-	0.1	-	0.1	V
		I_O = 4.0 mA; V_{CC} = 4.5 V	-	0.15	0.26	-	0.33	-	0.4	V
		I_O = 5.2 mA; V_{CC} = 6.0 V	-	0.16	0.26	-	0.33	-	0.4	V
I_I	input leakage current	$V_I = V_{CC}$ or GND; V_{CC} = 6.0 V	-	-	±0.1	-	±1.0	-	±1.0	µA
I_{CC}	supply current	$V_I = V_{CC}$ or GND; I_O = 0 A; V_{CC} = 6.0 V	-	-	2.0	-	20	-	40	µA
C_I	input capacitance		-	3.5	-	-	-	-	-	pF
74HCT14										
V_{OH}	HIGH-level output voltage	$V_I = V_{T+}$ or V_{T-}; V_{CC} = 4.5 V								
		$I_O = -20$ µA	4.4	4.5	-	4.4	-	4.4	-	V
		$I_O = -4.0$ mA	3.98	4.32	-	3.84	-	3.7	-	V
V_{OL}	LOW-level output voltage	$V_I = V_{T+}$ or V_{T-}; V_{CC} = 4.5 V								
		I_O = 20 µA;	-	0	0.1	-	0.1	-	0.1	V
		I_O = 4.0 mA;	-	0.15	0.26	-	0.33	-	0.4	V
I_I	input leakage current	$V_I = V_{CC}$ or GND; V_{CC} = 5.5 V	-	-	±0.1	-	±1.0	-	±1.0	µA
I_{CC}	supply current	$V_I = V_{CC}$ or GND; I_O = 0 A; V_{CC} = 5.5 V	-	-	2.0	-	20	-	40	µA
ΔI_{CC}	additional supply current	per input pin; $V_I = V_{CC} - 2.1$ V; other pins at V_{CC} or GND; I_O = 0 A; V_{CC} = 4.5 V to 5.5 V	-	30	108	-	135	-	147	µA
C_I	input capacitance		-	3.5	-	-	-	-	-	pF

11. Dynamic characteristics

Table 7. Dynamic characteristics
GND = 0 V; C_L = 50 pF; for test circuit see Figure 7.

Symbol	Parameter	Conditions		T_{amb} = 25 °C			T_{amb} = −40 °C to +125 °C		Unit
				Min	Typ	Max	Max (85 °C)	Max (125 °C)	
74HC14									
t_{pd}	propagation delay	nA to nY; see Figure 6	[1]						
		V_{CC} = 2.0 V		-	41	125	155	190	ns
		V_{CC} = 4.5 V		-	15	25	31	38	ns
		V_{CC} = 5.0 V; C_L = 15 pF		-	12	-	-	-	ns
		V_{CC} = 6.0 V		-	12	21	26	32	ns
t_t	transition time	see Figure 6	[2]						
		V_{CC} = 2.0 V		-	19	75	95	110	ns
		V_{CC} = 4.5 V		-	7	15	19	22	ns
		V_{CC} = 6.0 V		-	6	13	15	19	ns
C_{PD}	power dissipation capacitance	per package; V_I = GND to V_{CC}	[3]	-	7	-	-	-	pF
74HCT14									
t_{pd}	propagation delay	nA to nY; see Figure 6	[1]						
		V_{CC} = 4.5 V		-	20	34	43	51	ns
		V_{CC} = 5.0 V; C_L = 15 pF		-	17	-	-	-	ns
t_t	transition time	V_{CC} = 4.5 V; see Figure 6	[2]	-	7	15	19	22	ns
C_{PD}	power dissipation capacitance	per package; V_I = GND to V_{CC} − 1.5 V	[3]	-	8	-	-	-	pF

[1] t_{pd} is the same as t_{PHL} and t_{PLH}.

[2] t_t is the same as t_{THL} and t_{TLH}.

[3] C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$P_D = C_{PD} \times V_{CC}^2 \times f_i \times N + \Sigma (C_L \times V_{CC}^2 \times f_o)$ where:

f_i = input frequency in MHz;

f_o = output frequency in MHz;

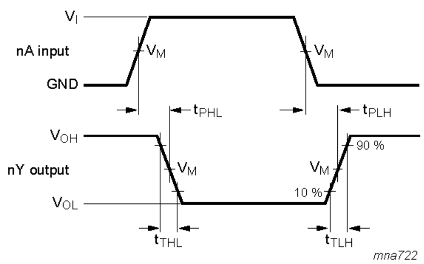
C_L = output load capacitance in pF;

V_{CC} = supply voltage in V;

N = number of inputs switching;

$\Sigma (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs.

12. Waveforms



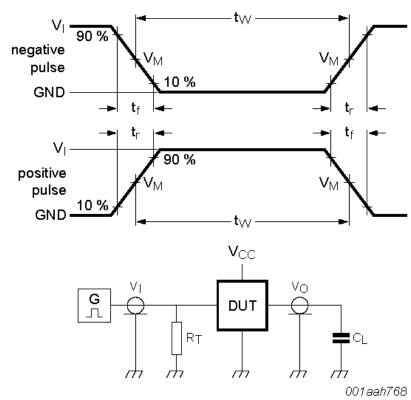
Measurement points are given in Table 8.

V_{OL} and V_{OH} are typical voltage output levels that occur with the output load.

Fig 6. Input to output propagation delays

Table 8. Measurement points

Type	Input	Output		
	V_M	V_M	V_X	V_Y
74HC14	$0.5V_{CC}$	$0.5V_{CC}$	$0.1V_{CC}$	$0.9V_{CC}$
74HCT14	1.3 V	1.3 V	$0.1V_{CC}$	$0.9V_{CC}$



Test data is given in Table 9.

Definitions test circuit:

R_T = termination resistance should be equal to output impedance Z_o of the pulse generator.

C_L = load capacitance including jig and probe capacitance.

Fig 7. Load circuitry for measuring switching times

74HC14; 74HCT14

Hex inverting Schmitt trigger

Rev. 6 — 19 September 2012 **Product data sheet**

1. General description

The 74HC14; 74HCT14 is a high-speed Si-gate CMOS device and is pin compatible with Low-power Schottky TTL (LSTTL). It is specified in compliance with JEDEC standard No. 7A.

The 74HC14; 74HCT14 provides six inverting buffers with Schmitt-trigger action. It is capable of transforming slowly changing input signals into sharply defined, jitter-free output signals.

2. Features and benefits

- Low-power dissipation
- ESD protection:
  - HBM JESD22-A114F exceeds 2000 V
  - MM JESD22-A115-A exceeds 200 V
- Multiple package options
- Specified from −40 °C to +85 °C and from −40 °C to +125 °C

3. Applications

- Wave and pulse shapers
- Astable multivibrators
- Monostable multivibrators

4. Ordering information

Table 1. Ordering information

Type number	Package			
	Temperature range	Name	Description	Version
74HC14N 74HCT14N	−40 °C to +125 °C	DIP14	plastic dual in-line package; 14 leads (300 mil)	SOT27-1
74HC14D 74HCT14D	−40 °C to +125 °C	SO14	plastic small outline package; 14 leads; body width 3.9 mm	SOT108-1
74HC14DB 74HCT14DB	−40 °C to +125 °C	SSOP14	plastic shrink small outline package; 14 leads; body width 5.3 mm	SOT337-1
74HC14PW 74HCT14PW	−40 °C to +125 °C	TSSOP14	plastic thin shrink small outline package; 14 leads; body width 4.4 mm	SOT402-1
74HC14BQ 74HCT14BQ	−40 °C to +125 °C	DHVQFN14	plastic dual in-line compatible thermal enhanced very thin quad flat package; no leads; 14 terminals; body 2.5 × 3 × 0.85 mm	SOT762-1

5. Functional diagram

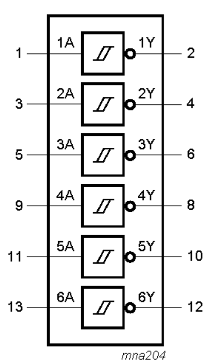


Fig 1. Logic symbol

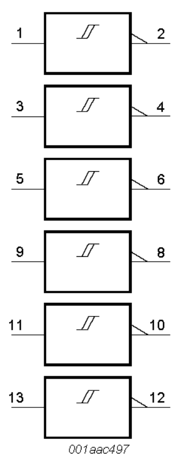


Fig 2. IEC logic symbol

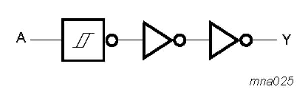


Fig 3. Logic diagram (one Schmitt trigger)

6. Pinning information

6.1 Pinning

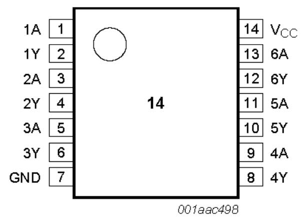
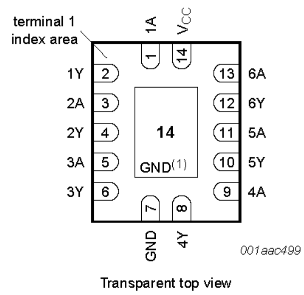


Fig 4. Pin configuration DIP14, SO14 and (T)SSOP14



(1) The die substrate is attached to this pad using conductive die attach material. It cannot be used as a supply pin or input.

Fig 5. Pin configuration DHVQFN14

6.2 Pin description

Table 2. Pin description

Symbol	Pin	Description
1A to 6A	1, 3, 5, 9, 11, 13	data input 1
1Y to 6Y	2, 4, 6, 8, 10, 12	data output 1
GND	7	ground (0 V)
V_{CC}	14	supply voltage

7. Functional description

Table 3. Function table[1]

Input	Output
nA	nY
L	H
H	L

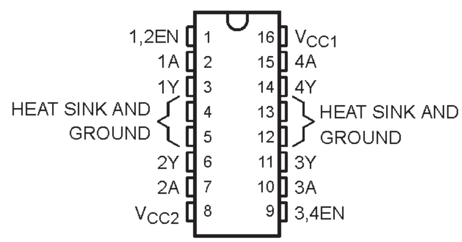
[1] H = HIGH voltage level;
L = LOW voltage level.

SN754410
QUADRUPLE HALF-H DRIVER

SLRS007B – NOVEMBER 1986 – REVISED NOVEMBER 1995

- 1-A Output-Current Capability Per Driver
- Applications Include Half-H and Full-H Solenoid Drivers and Motor Drivers
- Designed for Positive-Supply Applications
- Wide Supply-Voltage Range of 4.5 V to 36 V
- TTL- and CMOS-Compatible High-Impedance Diode-Clamped Inputs
- Separate Input-Logic Supply
- Thermal Shutdown
- Internal ESD Protection
- Input Hysteresis Improves Noise Immunity
- 3-State Outputs
- Minimized Power Dissipation
- Sink/Source Interlock Circuitry Prevents Simultaneous Conduction
- No Output Glitch During Power Up or Power Down
- Improved Functional Replacement for the SGS L293

NE PACKAGE
(TOP VIEW)



FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level
X = irrelevant
Z = high-impedance (off)
† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

description

The SN754410 is a quadruple high-current half-H driver designed to provide bidirectional drive currents up to 1 A at voltages from 4.5 V to 36 V. The device is designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are compatible with TTL-and low-level CMOS logic. Each output (Y) is a complete totem-pole driver with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs become active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in a high-impedance state. With the proper data inputs, each pair of drivers form a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

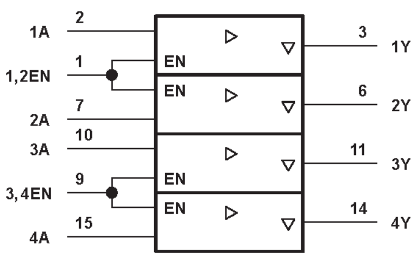
A separate supply voltage (V_{CC1}) is provided for the logic input circuits to minimize device power dissipation. Supply voltage V_{CC2} is used for the output circuits.

The SN754410 is designed for operation from −40°C to 85°C.

SN754410
QUADRUPLE HALF-H DRIVER

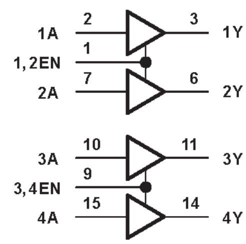
SLRS007B – NOVEMBER 1986 – REVISED NOVEMBER 1995

logic symbol†

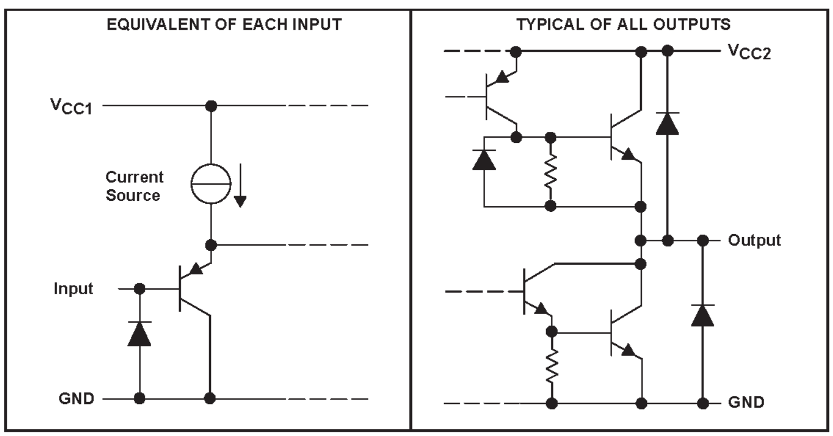


† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

logic diagram



schematics of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Output supply voltage range, V_{CC1} (see Note 1)	−0.5 V to 36 V
Output supply voltage range, V_{CC2}	−0.5 V to 36 V
Input voltage, V_I	36 V
Output voltage range, V_O	−3 V to V_{CC2} + 3 V
Peak output current (nonrepetitive, $t_w \le 5$ ms)	±2 A
Continuous output current, I_O	±1.1 A
Continuous total power dissipation at (or below) 25°C free-air temperature (see Note 2)	2075 mW
Operating free-air temperature range, T_A	−40°C to 85°C
Operating virtual junction temperature range, T_J	−40°C to 150°C
Storage temperature range, T_{stg}	−65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to network GND.
2. For operation above 25°C free-air temperature, derate linearly at the rate of 16.6 mW/°C. To avoid exceeding the design maximum virtual junction temperature, these ratings should not be exceeded. Due to variations in individual device electrical characteristics and thermal resistance, the built-in thermal overload protection can be activated at power levels slightly above or below the rated dissipation.

recommended operating conditions

	MIN	MAX	UNIT
Output supply voltage, V_{CC1}	4.5	5.5	V
Output supply voltage, V_{CC2}	4.5	36	V
High-level input voltage, V_{IH}	2	5.5	V
Low-level input voltage, V_{IL}	−0.3‡	0.8	V
Operating virtual junction temperature, T_J	−40	125	°C
Operating free-air temperature, T_A	−40	85	°C

‡ The algebraic convention, in which the least positive (most negative) limit is designated as minimum, is used in this data sheet for logic voltage levels.

electrical characteristics over recommended ranges of supply voltage and free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{IK}	Input clamp voltage	$I_I = -12$ mA		−0.9	−1.5	V
V_{OH}	High-level output voltage	$I_{OH} = -0.5$ A	$V_{CC2}-1.5$	$V_{CC2}-1.1$		V
		$I_{OH} = -1$ A	$V_{CC2}-2$			
		$I_{OH} = -1$ A, $T_J = 25°C$	$V_{CC2}-1.8$	$V_{CC2}-1.4$		
V_{OL}	Low-level output voltage	$I_{OL} = 0.5$ A		1	1.4	V
		$I_{OL} = 1$ A			2	
		$I_{OL} = 1$ A, $T_J = 25°C$		1.2	1.8	
V_{OKH}	High-level output clamp voltage	$I_{OK} = -0.5$ A		$V_{CC2}+1.4$	$V_{CC2}+2$	V
		$I_{OK} = 1$ A		$V_{CC2}+1.9$	$V_{CC2}+2.5$	
V_{OKL}	Low-level output clamp voltage	$I_{OK} = 0.5$ A		−1.1	−2	V
		$I_{OK} = -1$ A		−1.3	−2.5	
$I_{OZ(off)}$	Off-state high-impedance-state output current	$V_O = V_{CC2}$			500	µA
		$V_O = 0$			−500	
I_{IH}	High-level input current	$V_I = 5.5$ V			10	µA
I_{IL}	Low-level input current	$V_I = 0$			−10	µA
I_{CC1}	Output supply current	$I_O = 0$, All outputs at high level			38	mA
		$I_O = 0$, All outputs at low level			70	
		$I_O = 0$, All outputs at high impedance			25	
I_{CC2}	Output supply current	$I_O = 0$, All outputs at high level			33	mA
		$I_O = 0$, All outputs at low level			20	
		$I_O = 0$, All outputs at high impedance			5	

† All typical values are at $V_{CC1} = 5$ V, $V_{CC2} = 24$ V, $T_A = 25°C$.

switching characteristics, $V_{CC1} = 5$ V, $V_{CC2} = 24$ V, $C_L = 30$ pF, $T_A = 25°C$

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time, high-to-low-level output from A input	See Figure 1		400		ns
t_{d2}	Delay time, low-to-high-level output from A input			800		ns
t_{TLH}	Transition time, low-to-high-level output			300		ns
t_{THL}	Transition time, high-to-low-level output			300		ns
t_r	Rise time, pulse input					
t_f	Fall time, pulse input					
t_w	Pulse duration					
t_{en1}	Enable time to the high level	See Figure 2		700		ns
t_{en2}	Enable time to the low level			400		ns
t_{dis1}	Disable time from the high level			900		ns
t_{dis2}	Disable time from the low level			600		ns

PARAMETER MEASUREMENT INFORMATION

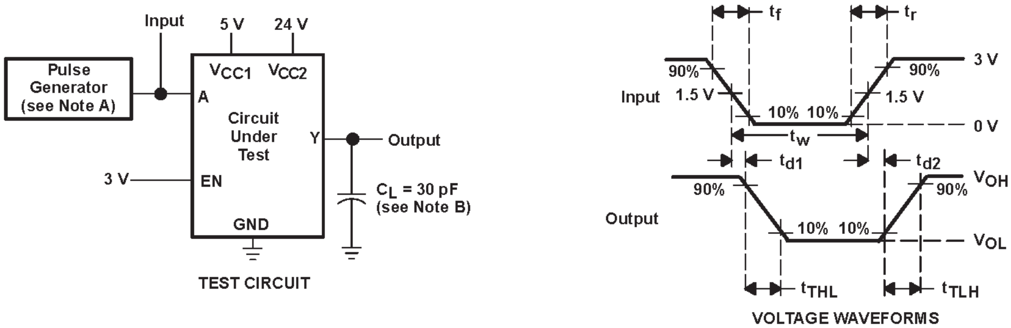


Figure 1. Test Circuit and Switching Times From Data Inputs

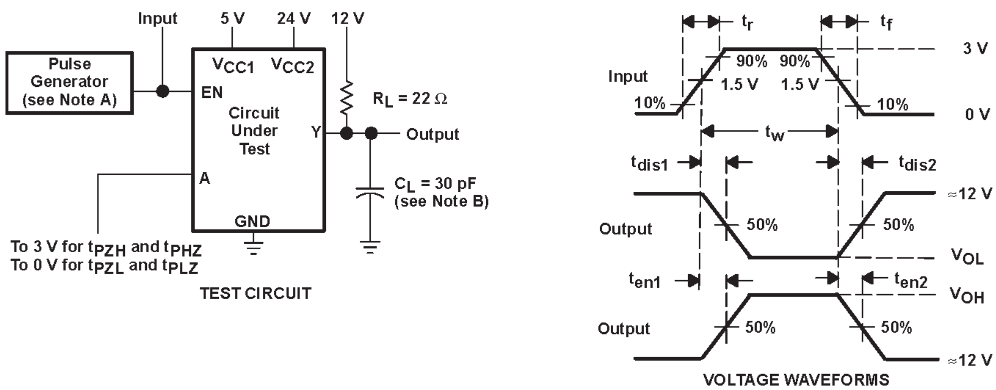


Figure 2. Test Circuit and Switching Times From Enable Inputs

NOTES: A. The pulse generator has the following characteristics: $t_r \leq 10$ ns, $t_f \leq 10$ ns, $t_w = 10$ µs, PRR = 5 kHz, $Z_O = 50\ \Omega$.
B. C_L includes probe and jig capacitance.

SN754410
QUADRUPLE HALF-H DRIVER

APPLICATION INFORMATION

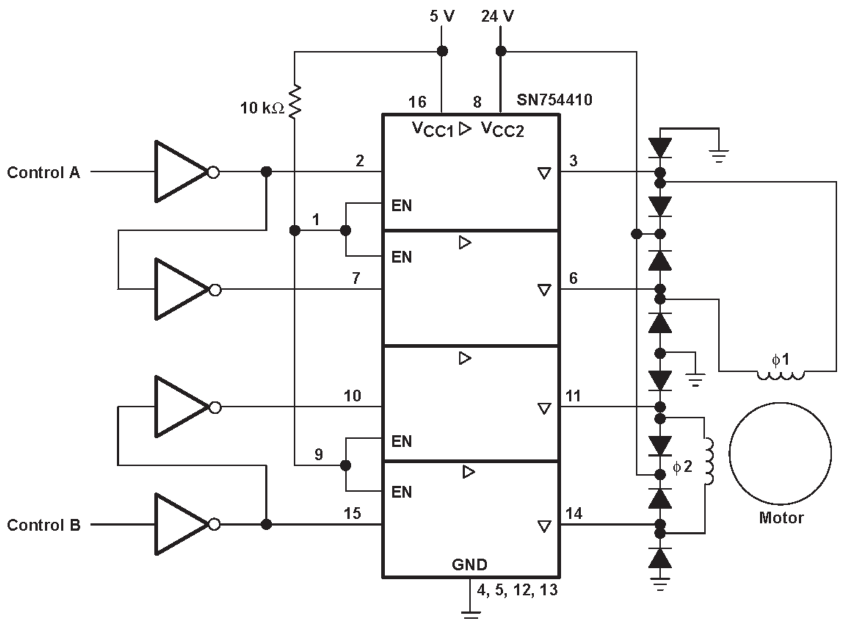


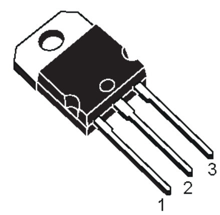
Figure 3. Two-Phase Motor Driver

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 ● DALLAS, TEXAS 75265
POST OFFICE BOX 1443 ● HOUSTON, TEXAS 77251–1443

TIP140/141/142
TIP145/146/147

COMPLEMENTARY SILICON POWER DARLINGTON TRANSISTORS

- TIP141, TIP142, TIP145 AND TIP147 ARE STMicroelectronics PREFERRED SALESTYPES
- COMPLEMENTARY PNP - NPN DEVICES
- MONOLITHIC DARLINGTON CONFIGURATION
- INTEGRATED ANTIPARALLEL COLLECTOR-EMITTER DIODE



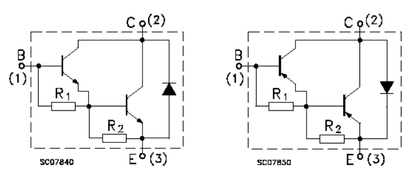
TO-218

APPLICATIONS
- LINEAR AND SWITCHING INDUSTRIAL EQUIPMENT

DESCRIPTION
The TIP140, TIP141 and TIP142 are silicon Epitaxial-Base NPN power transistors in monolithic Darlington configuration, mounted in TO-218 plastic package. They are intented for use in power linear and switching applications.

The complementary PNP types are TIP145, TIP146 and TIP147 respectively.

INTERNAL SCHEMATIC DIAGRAM



R_1 Typ. = 5 KΩ R_2 Typ. = 150 Ω

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter		Value			Unit
		NPN	TIP140	TIP141	TIP142	
		PNP	TIP145	TIP146	TIP147	
V_{CBO}	Collector-Base Voltage (I_E = 0)		60	80	100	V
V_{CEO}	Collector-Emitter Voltage (I_B = 0)		60	80	100	V
V_{EBO}	Emitter-Base Voltage (I_C = 0)			5		V
I_C	Collector Current			10		A
I_{CM}	Collector Peak Current			20		A
I_B	Base Current			0.5		A
P_{tot}	Total Dissipation at $T_{case} \leq 25$ °C			125		W
T_{stg}	Storage Temperature			-65 to 150		°C
T_j	Max. Operating Junction Temperature			150		°C

For PNP types voltage and current values are negative.

THERMAL DATA

$R_{thj-case}$	Thermal Resistance Junction-case	Max	1	°C/W

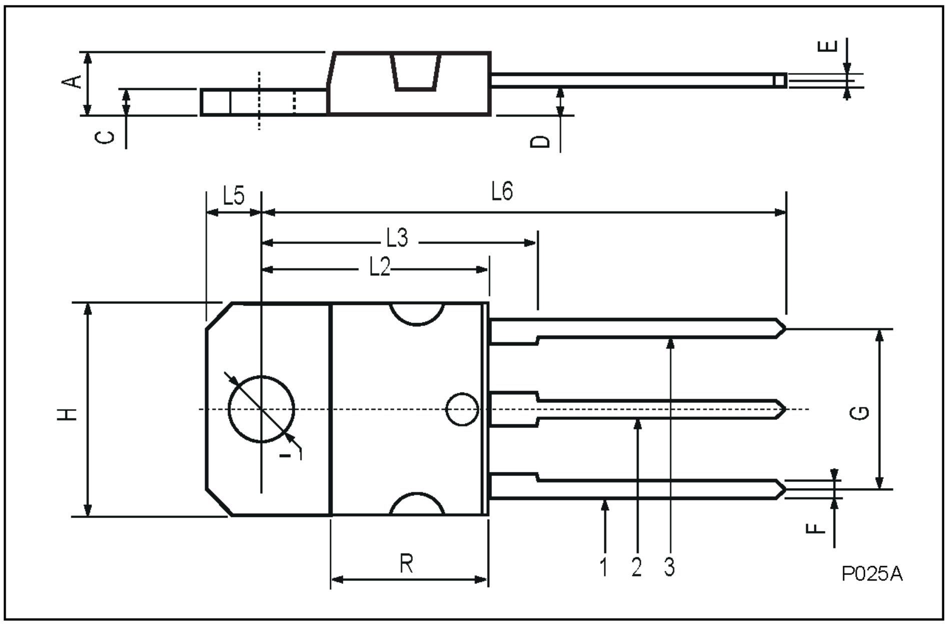
ELECTRICAL CHARACTERISTICS (T_{case} = 25 °C unless otherwise specified)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
I_{CBO}	Collector Cut-off Current (I_E = 0)	for **TIP140/145** for **TIP141/146** for **TIP142/147**	V_{CB} = 60 V V_{CB} = 80 V V_{CB} = 100 V			1 1 1	mA mA mA
I_{CEO}	Collector Cut-off Current (I_B = 0)	for **TIP140/145** for **TIP141/146** for **TIP142/147**	V_{CE} = 30 V V_{CE} = 40 V V_{CE} = 50 V			2 2 2	mA mA mA
I_{EBO}	Emitter Cut-off Current (I_C = 0)	V_{EB} = 5 V				2	mA
$V_{CEO(sus)}$*	Collector-Emitter Sustaining Voltage (I_B = 0)	I_C = 30 mA for **TIP140/145** for **TIP141/146** for **TIP142/147**		 60 80 100			 V V V
$V_{CE(sat)}$*	Collector-Emitter Saturation Voltage	I_C = 5 A I_C = 10 A	I_B = 10 mA I_B = 40 mA			2 3	V V
$V_{BE(on)}$*	Base-Emitter Voltage	I_C = 10 A	V_{CE} = 4 V			3	V
h_{FE}*	DC Current Gain	I_C = 5 A I_C = 10 A	V_{CE} = 4 V V_{CE} = 4 V	1000 500			
 t_{on} t_{off}	RESISTIVE LOAD Turn-on Time Turn-off Time	I_C = 10 A I_{B2} = -40 mA	I_{B1} = 40 mA R_L = 3 Ω		 0.9 4		 μs μs

For PNP types voltage and current values are negative.
∗ Pulsed: Pulse duration = 300 μs, duty cycle 1.5 %

TO-218 (SOT-93) MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	4.7		4.9	0.185		0.193
C	1.17		1.37	0.046		0.054
D		2.5			0.098	
E	0.5		0.78	0.019		0.030
F	1.1		1.3	0.043		0.051
G	10.8		11.1	0.425		0.437
H	14.7		15.2	0.578		0.598
L2	–		16.2	–		0.637
L3		18			0.708	
L5	3.95		4.15	0.155		0.163
L6		31			1.220	
R	–		12.2	–		0.480
Ø	4		4.1	0.157		0.161



FAIRCHILD
SEMICONDUCTOR™

October 1987
Revised April 2001

MM74C922 • MM74C923
16-Key Encoder • 20-Key Encoder

General Description

The MM74C922 and MM74C923 CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal debounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two-key roll-over is provided between any two switches.

An internal register remembers the last key pressed even after the key is released. The 3-STATE outputs provide for easy expansion and bus operation and are LPTTL compatible.

Features

- 50 kΩ maximum switch on resistance
- On or off chip clock
- On-chip row pull-up devices
- 2 key roll-over
- Keybounce elimination with single capacitor
- Last key register at outputs
- 3-STATE output LPTTL compatible
- Wide supply range: 3V to 15V
- Low power consumption

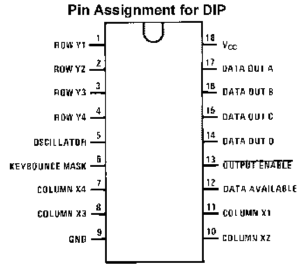
Ordering Code:

Order Number	Package Number	Package Description
MM74C922WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C922N	N18B	18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74C923WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Device also available in Tape and Reel. Specify by appending suffix letter "X" to the ordering code.

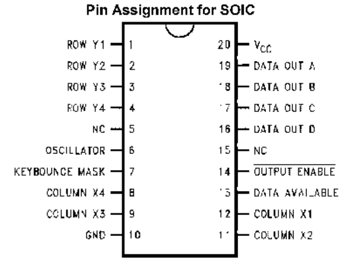
Connection Diagrams

Pin Assignment for DIP



Top View
MM74C922

Pin Assignment for SOIC

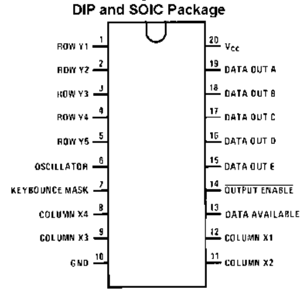


Top View
MM74C922

 DS006037 www.fairchildsemi.com

Connection Diagrams (Continued)

Pin Assignment for
DIP and SOIC Package



Top View
MM74C923

Truth Tables

(Pins 0 through 11)

Switch Position		0	1	2	3	4	5	6	7	8	9	10	11
		Y1, X1	Y1, X2	Y1, X3	Y1, X4	Y2, X1	Y2, X2	Y2, X3	Y2, X4	Y3, X1	Y3, X2	Y3, X3	Y3, X4
D													
A	A	0	1	0	1	0	1	0	1	0	1	0	1
T	B	0	0	1	1	0	0	1	1	0	0	1	1
A	C	0	0	0	0	1	1	1	1	0	0	0	0
O	D	0	0	0	0	0	0	0	0	1	1	1	1
U	E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0
T													

(Pins 12 through 19)

Switch Position		12	13	14	15	16	17	18	19
		Y4, X1	Y4, X2	Y4, X3	Y4, X4	Y5(Note 1), X1	Y5 (Note 1), X2	Y5 (Note 1), X3	Y5 (Note 1), X4
D									
A	A	0	1	0	1	0	1	0	1
T	B	0	0	1	1	0	0	1	1
A	C	1	1	1	1	0	0	0	0
O	D	1	1	1	1	0	0	0	0
U	E (Note 1)	0	0	0	0	1	1	1	1
T									

Note 1: Omit for MM74C922

HD44780U (LCD-II)

(Dot Matrix Liquid Crystal Display Controller/Driver)

HITACHI

ADE-207-272(Z)
'99.9
Rev. 0.0

Description

The HD44780U dot-matrix liquid crystal display controller and driver LSI displays alphanumerics, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

A single HD44780U can display up to one 8-character line or two 8-character lines.

The HD44780U has pin function compatibility with the HD44780S which allows the user to easily replace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 208 5 × 8 dot character fonts and 32 5 × 10 dot character fonts for a total of 240 different character fonts.

The low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-driven product requiring low power dissipation.

Features

- 5 × 8 and 5 × 10 dot matrix possible
- Low power operation support:
  — 2.7 to 5.5V
- Wide range of liquid crystal display driver power
  — 3.0 to 11V
- Liquid crystal drive waveform
  — A (One line frequency AC waveform)
- Correspond to high speed MPU bus interface
  — 2 MHz (when V_{CC} = 5V)
- 4-bit or 8-bit MPU interface enabled
- 80 × 8-bit display RAM (80 characters max.)
- 9,920-bit character generator ROM for a total of 240 character fonts
  — 208 character fonts (5 × 8 dot)
  — 32 character fonts (5 × 10 dot)

HD44780U

- 64 × 8-bit character generator RAM
  - — 8 character fonts (5 × 8 dot)
  - — 4 character fonts (5 × 10 dot)
- 16-common × 40-segment liquid crystal display driver
- Programmable duty cycles
  - — 1/8 for one line of 5 × 8 dots with cursor
  - — 1/11 for one line of 5 × 10 dots with cursor
  - — 1/16 for two lines of 5 × 8 dots with cursor
- Wide range of instruction functions:
  - — Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- Pin function compatibility with HD44780S
- Automatic reset circuit that initializes the controller/driver after power on
- Internal oscillator with external resistors
- Low power consumption

Ordering Information

Type No.	Package	CGROM
HD44780UA00FS	FP-80B	Japanese standard font
HCD44780UA00	Chip	
HD44780UA00TF	TFP-80F	
HD44780UA02FS	FP-80B	European standard font
HCD44780UA02	Chip	
HD44780UA02TF	TFP-80F	
HD44780UBxxFS	FP-80B	Custom font
HCD44780UBxx	Chip	
HD44780UBxxTF	TFP-80F	

Note: xx: ROM code No.

Pin Functions

Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/$\overline{W}$	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
SEG1 to SEG40	40	O	LCD	Segment signals
V1 to V5	5	—	Power supply	Power supply for LCD drive V_{CC} –V5 = 11 V (max)
V_{CC}, GND	2	—	Power supply	V_{CC}: 2.7V to 5.5V, GND: 0V
OSC1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.

Function Description

Registers

The HD44780U has two 8-bit registers, an instruction register (IR) and a data register (DR).

The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/$\overline{W}$ = 1 (Table 1), the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)

The address counter (AC) assigns addresses to both DDRAM and CGRAM. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when RS = 0 and R/$\overline{W}$ = 1 (Table 1).

Table 1 Register Selection

RS	R/$\overline{W}$	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Display Data RAM (DDRAM)

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 × 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.

- 1-line display (N = 0) (Figure 2)
  — When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.
  When the display shift operation is performed, the DDRAM address shifts. See Figure 3.

	High order bits			Low order bits			
AC (hexadecimal)	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Example: DDRAM address 4E

1	0	0	1	1	1	0

Figure 1 DDRAM Address

Display position (digit)	1	2	3	4	5		79	80
DDRAM address (hexadecimal)	00	01	02	03	04	....	4E	4F

Figure 2 1-Line Display

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
For shift left	01	02	03	04	05	06	07	08
For shift right	4F	00	01	02	03	04	05	06

Figure 3 1-Line by 8-Character Display Example

- 2-line display (N = 1) (Figure 4)
  - — Case 1: When the number of display characters is less than 40 × 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters × 2 lines are displayed. See Figure 5.

    When display shift operation is performed, the DDRAM address shifts. See Figure 5.

Display position	1	2	3	4	5		39	40
DDRAM address (hexadecimal)	00	01	02	03	04	....	26	27
	40	41	42	43	44	....	66	67

Figure 4 2-Line Display

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
	40	41	42	43	44	45	46	47
For shift left	01	02	03	04	05	06	07	08
	41	42	43	44	45	46	47	48
For shift right	27	00	01	02	03	04	05	06
	67	40	41	42	43	44	45	46

Figure 5 2-Line by 8-Character Display Example

— Case 2: For a 16-character × 2-line display, the HD44780 can be extended using one 40-output extension driver. See Figure 6.

When display shift operation is performed, the DDRAM address shifts. See Figure 6.

Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

$\underbrace{\text{positions 1–8}}_{\text{HD44780U display}}$ $\underbrace{\text{positions 9–16}}_{\text{Extension driver display}}$

For shift left	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50
For shift right	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

Figure 6 2-Line by 16-Character Display Example

HD44780U

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	β	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	×	万
xxxx1100	(5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)		-	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	゜	ö	█

Note: The user can specify any pattern for character-generator RAM.

HD44780U

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A02)

18 HITACHI

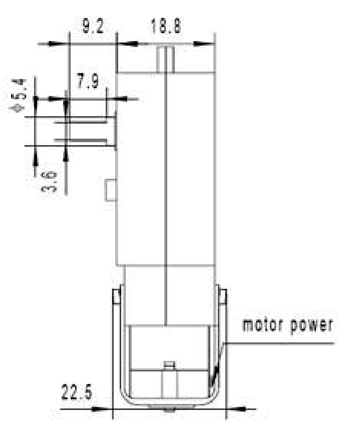
DC Characteristics (V_{CC} = 2.7 to 4.5 V, T_a = –30 to +75°C*[3])

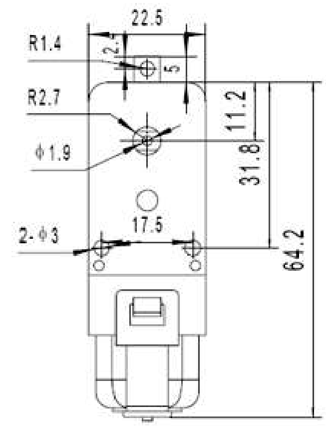
Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	$0.7V_{CC}$	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	–0.3	—	0.55	V		6
Input high voltage (2) (OSC1)	VIH2	$0.7V_{CC}$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	$0.2V_{CC}$	V		15
Output high voltage (1) (DB0–DB7)	VOH1	$0.75V_{CC}$	—	—	V	$-I_{OH}$ = 0.1 mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	$0.2V_{CC}$	V	I_{OL} = 0.1 mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	$0.8V_{CC}$	—	—	V	$-I_{OH}$ = 0.04 mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	$0.2V_{CC}$	V	I_{OL} = 0.04 mA	8
Driver on resistance (COM)	R_{COM}	—	2	20	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Driver on resistance (SEG)	R_{SEG}	—	2	30	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Input leakage current	I_{LI}	–1	—	1	µA	VIN = 0 to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/$\overline{W}$)	$-I_p$	10	50	120	µA	V_{CC} = 3 V	
Power supply current	I_{CC}	—	150	300	µA	R_f oscillation, external clock V_{CC} = 3 V, f_{OSC} = 270 kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	V_{CC}–V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC}–V5, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Model：TGP01S-A130
Gear Motor

Applications
Description
Reduction Ratio:
1/48,、1/120、1/180、1/220、1/256、1/288





Weight: 35g Unit: mm

MODEL		VOLTAGE		NO LOAD		AT LOAD					STALL		
		OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			V	rpm	A	rpm	A	N.m	Kg.cm	W	N.m	Kg.cm	A
TGP01S-A130	14150-120	3.0-12.0	4.5	100	0.20	87	0.50	0.066	0.675	0.40	0.265	2.700	1.25
	18100-220	3.0-12.0	3.0	50	0.20	40	0.60	0.120	1.233	0.6	0.393	4.003	2.15

The product you have searched for is available at the ELECTRONIC GOLDMINE store.

Powerful 12VDC to 24VDC Spring Return Solenoid

Item Number : G16036
Unit Price: 0.99

Detailed Description

This is a powerful pull-type solenoid model CII / A1464. Features all metal construction and operates from 12VDC to 24VDC. Coil resistance is 3.3 to 4.2 ohms. At 24VDC, the pull force is 3.7 lbs. Metal plunger is 3” long x .37” Diameter. Body is 2.15” Long x 0.77” Diameter and is attached to a 3.6” Long metal mounting bracket.
G16036