UNIVERSITY OF TORONTO

INSTITUTE FOR AEROSPACE STUDIES

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

EcoSort
Bottle Sorting Machine

prepared by

Team 52 – Tuesday

Maliha Islam (1002262295)
Rosemary Jose (1002196189)
Jeffrey Qiu (1002261562)

prepared for

Prof. M.R. Emami

A technical report submitted for
AER201 – Engineering Design

TA: Thomas Ulph

EcoSort

Bottle Recycling Machine

Course Instructor: Dr. Reza Emami

Teaching Assistant: Thomas Ulph

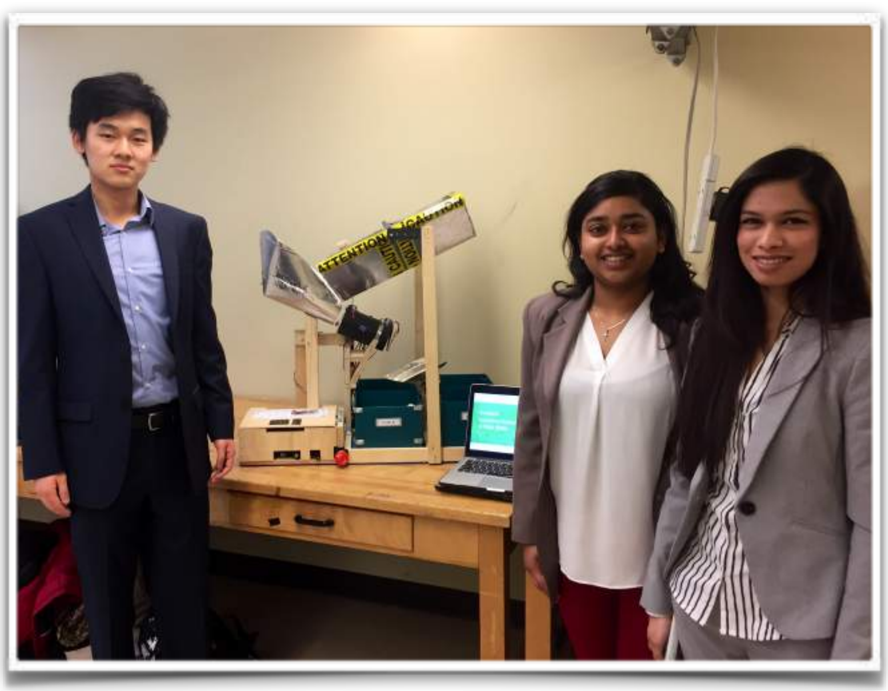
AER201: Engineering Design

Team 52

Maliha Islam | 1002262295

Rosemary Jose | 1002196189

Jeffrey Qiu | 1002261562



Division of Engineering Science, University of Toronto

April 13, 2017

ACKNOWLEDGEMENTS

We would like to acknowledge Professor Reza Emami, the teaching team and the Division of Engineering Science for providing us with the opportunity to develop hands-on engineering and design skills through the AER201: Engineering Design course.

Special thanks to our teaching assistant, Thomas Ulph, and the shop teaching assistant, Colin Harry, who provided exceptional design support and guidance which were vital to the success of this project.

We would also like to extend our gratitude to all our colleagues in the class of 1T9; your support, encouragement and camaraderie has been key to the successful completion of our AER201 project.

Our gratitude also goes out to all our peers in the Engineering Science program who have tolerated our overwhelming presence in the Engineering Science Common Room and encouraged our success in the AER201.

ABSTRACT

The request for proposal (RFP) given asks teams to create an autonomous bottle sorting robot that can distinguish between four types of bottles based on their opacity and whether they have caps. Such a machine would improve the efficiency and effectiveness of recycling plants, by automating what is usually tedious, manual labour.

The machine discussed in this report, EcoSort has three key mechanisms: loading and orientation, identification, and, delivery. Upon activation, the V-shaped loading cradle rocks up and down agitating bottles and allowing them to flow in single stream into the testing chamber. In the chamber, a series of sensors identify the bottle. Finally, the delivery chute transports the bottles to the accurate delivery container; the results of the sorting and the time of operation are displayed on an LCD screen.

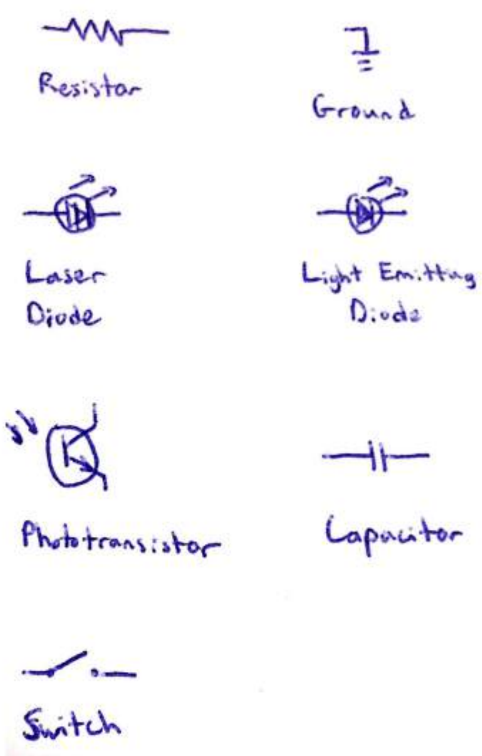
This report also discusses each subsystem in great detail. Further, it includes background information, budget, proposed and actual timelines, areas for improvement and operation protocols.

The performance of the AER201 team is also evaluated with a holistic, practical perspective as introduced in the Engineering Design course lectures.

TABLE OF CONTENTS

1. SYMBOLS AND ABBREVIATIONS

1.1 Notation



Symbol	Meaning	Units
V	Voltage	[V]
I	Current	[A]
P	Power	[W]
R	Resistance	[Ω], Ohms
τ	Torque	[Nm]
F	Force	[N]
r	Distance	[m]
I	Moment of Inertia	[kg m^2]

Symbol	Meaning	Units
α	Angular Acceleration	[rad/s^2]

Table 1. Symbols Used In Calculations

1.2 Abbreviations

Meaning	Meaning
ANSI C	American National Standards Institute C Programming
CAD	Canadian Dollar
DC	Direct Current
HDPE	High Density PolyEthylene
I/O	Input/ Output
IDE	Integrated Development Environment
IR LED	InfraRed Light Emitting Diode
LCD	Liquid Crystal Display
LTE	Long Term Evaluation
MCU	Microcontroller Unit
PET	PolyEthylene Terephthalate
PIC	Peripheral Interface Controller
PSD	Position Sensitive Detector
RFP	Request for Proposal
RGB	Red Green Blue
RTC	Real Time Chip
RTOS	Real Time Operating System

Table 2. Abbreviations Used In This Report

2. INTRODUCTION

Plastic bottles are widely manufactured, widely used and are a major focus in the recycling industry. Before the plastics are broken down and reformed, different types of plastics must be separated. Manual separation of plastics is a time and labour-intensive process, so an autonomous solution is desired. Two examples of commonly used plastic bottles are 200 mL white HDPE (eg. Yop) and 330 mL transparent PET (eg. Eska).

This report discusses an autonomous machine that can sort two different types of plastic bottles -- Eska and Yop -- and differentiate between the bottles with and without caps.

The bottle recycling machine is decomposed based on function into three particular mechanisms:

1. Orientation, by means on a V-shaped cradle that rocks back and forth on its transverse axis
2. Identification, using proximity sensors placed in appropriate locations in a testing chamber
3. Delivery, by means of a rotating chute that deposits that sorted bottles into their respective containers

This report will first discuss the context and background of the project before delving in and presenting the design machine in terms of the three subsystems -- electromechanical, circuitry and sensors, and microcontroller. Then, integration of these three subsystems will be discussed based on the three mechanisms described above. Finally, budgeting, process, and potential overall improvements will finally

3. PROJECT CONCEPT

3.1 Statement of Need

A recycling company receives a continuous mixed stream of used plastic bottles of different types, and needs to sort them before sending them to the recycling plant. Manual sorting is both time and labour-intensive. In order to increase its effectiveness and efficiency, the recycling plant requests an autonomous machine that is capable of sorting plastic bottles of two types and of further sorting bottles with or without lids.

3.2 Goal

Design and fabricate an autonomous machine that is capable of identifying and sorting two different types of plastic bottles, as well as those bottles with or without lids.

3.3 Objective

Design and prototype a machine that can correctly identify and sort at least 10 plastic bottles of various types (Eska 330 ml or Yop 200 ml) and of various states (with lids or without lids) into separate delivery containers in 3 minutes or less.

3.4 Stakeholders

1. Recycling Plant (Client)
   Recycling plant clients are interested in correctly sorted plastic bottles so that each bottle is recycled using the bottle-specific method at an appropriate location. If incorrectly sorted bottles enter the wrong section of the plant, the recycling process will not be carried out in a proper and accurate manner.

2. Recycling Plant Employees (Users)
   Recycling plant employees are interested in a highly accurate and thus, reliable sorting machine since this will reduce manual labour of sorting the bottles or re-sorting bottles that were incorrectly sorted by the machine.

3. Proposed Machine Manufacturer
   The machine manufacturer will manufacture the machine parts and assemble them for market distribution.

4. Microcontroller Distributer
   The microcontroller distributer will provide the machine company with the PIC18F4620 microcontroller needed to automate the machine.

5. Design Teams Who Will Propose Solutions (eg: Team 52)
   The design team designing the machine is interested in meeting the needs of the recycling plant clients and other stakeholders, fulfilling the objectives and constraints outlined for the machine. The team is interested in delivering an accurate, fast and reliable machine to its clients.

3.5 Requirements

- Autonomous operation
- Portable with no need for installation
- Convenient and easy to use
- Must accept user's instructions through a keypad and LCD
- An easily accessible emergency STOP switch to cease all mechanical parts
- Must complete operations within 3 minutes or 180 seconds

3.6 Detailed Objectives, Constraints and Criteria [1]

Objectives	Constraints	Criteria
Size	Must not exceed 0.9 x 0.9 x 0.9 m^3	Smaller is better
Weight	Must not exceed 10kg	Lighter is better
Cost	Must not exceed CAD $230, before shipment and taxes	Cheaper is better, so long as other aspects of the design are not compromised
Autonomous	No interaction with PC or remote control during operation Must start on user input through keypad	Autonomy is preferred
Speed	Time required for loading the bottle shall not exceed 1 minute Sorting of one load (10 bottles) must not take longer than 3 minutes	Faster is better
Time	Little time is required to set up and calibrate the machine.	Less time is better

Table 3. Objectives, Constraints, And Criteria

[1] Refer to Appendix B to see the use theses criteria in our decision making

3.7 Team Design Values

- *Simplicity*
  The machine must have as few mechanical and electronic components as possible to reduce manufacturing and assembly complexity. The machine program must have low time complexity and run smoothly without any errors or bugs.

- *Convenience and Ease-of-Use*
  The machine user interface must be self-explanatory and provide easy navigation for users of various skill levels.

- *Operability and Sustainability*
  The machine set up and calibration must take little time and effort. Modularity is highly valued so that parts can be easily replaced or repaired.

- *Elegance and Safety*
  The machine must be aesthetically pleasing and operate quietly and smoothly with little or no sensible noise or vibration. It must have an easily accessible emergency STOP switch to stop all electrical moving mechanisms. The machine must not pose any hazard to the users and shall not be perceived as hazardous.

4. PERSPECTIVE: THEORY, HISTORY AND SURVEY

During the initial conceptual design phase, the project was divided to design components for three major sections, which are listed below in the order of the sorting process.

- *Loading and Orientation:* Designing an automated loading bin for 10 bottles which would align the bottles either horizontally or vertically after they are loaded in a random orientation
- *Identification:* Identifying and distinguishing between the four types of bottles: 200-ml white (HDPE) YOP brand bottles with or without cap, and 330-ml transparent (PTE) Eska brand bottles with or without cap
- *Sorting and Delivery:* Delivering the identified bottles to their respective containers; each of the four types has a designated, removable container

Although the three components depend on each other for full functionality of the machine, each component can be designed separately or changed without causing major disruptions to the system as a whole.

Reference designs are presented here.

4.1 Methods of Loading and Orientation

4.1.1 Titan Bottle Packing Line, Titan Pharmaceutical Machinery [2]

This industrial bottle sorting machine designed by Titan Pharmaceutical Machinery, uses a centrifugal loading bin that passes bottles horizontally through a spinning door onto a conveyor belt (Figure 1A), where a mechanical arm changes the format of bottles (Figure 1B). The arm is lowered onto the conveyor belt by default and catches a bottle's uncapped hollow mouth if the bottle is moving toward the arm with its mouth facing it and changes its format. Otherwise, the arm skips over the bottle. The finished result is horizontally aligned bottles with all their

Figure 1 Titan Bottle Packing Line With (A) Loading Bin And Conveyor Leading To (B) A Mechanical Arm

mouths facing away from and their rear ends facing the loading bin door. For the scope of the project, this design would require uncapping all the bottles first since at least some of the bottles are expected be capped.

4.1.2 Vision Guided Robotic Blow-Mold Take Away, Dyco Inc. [3]

This design, by Dyco Inc., accepts randomly oriented plastic containers from a blow molder and passes the containers on a conveyor belt where a Kuka 6-axis robotic arm locates and identifies the orientation of each container using camera vision. The robot picks it up and places the bottle upright (vertically) onto a single lane exit vacuum conveyor from which the containers are conveyed out in single file. However, since the conveyor belt keeps running, the robot misses some of the containers while it is picking up another container (Figure 2).

Figure 2 Vision Guided Robotic Arm Placing A Container On A Lane [3]

4.1.3 BCM Linear Unscrambler and Orienter Machine, BCM Engineering [4,5]

The Unscrambler consists of a vibrating table that orients the bottles horizontally by making them fit into the circular depressions (Figure 3A) and moves them into an orienting unit where they are placed in upright positions through tipping motions. The bottles are then released by separators towards a vacuum belt if the bottles have a stable shape (Figure 3B) or individual transport pucks if they have difficulties staying upright due to irregular shapes (Figure 3C). A transfer belt containing the bottles upright (vertically) from the Unscrambler Machine is fed into the Orienter Machine where they are spaced and separated by a transfer screw, detected by optical fibre or a camera and then rotated by a servo-motor 180º if necessary (Figure 3D). This system has many actuated components and intricate machinery, which is out of the scope of our objectives.

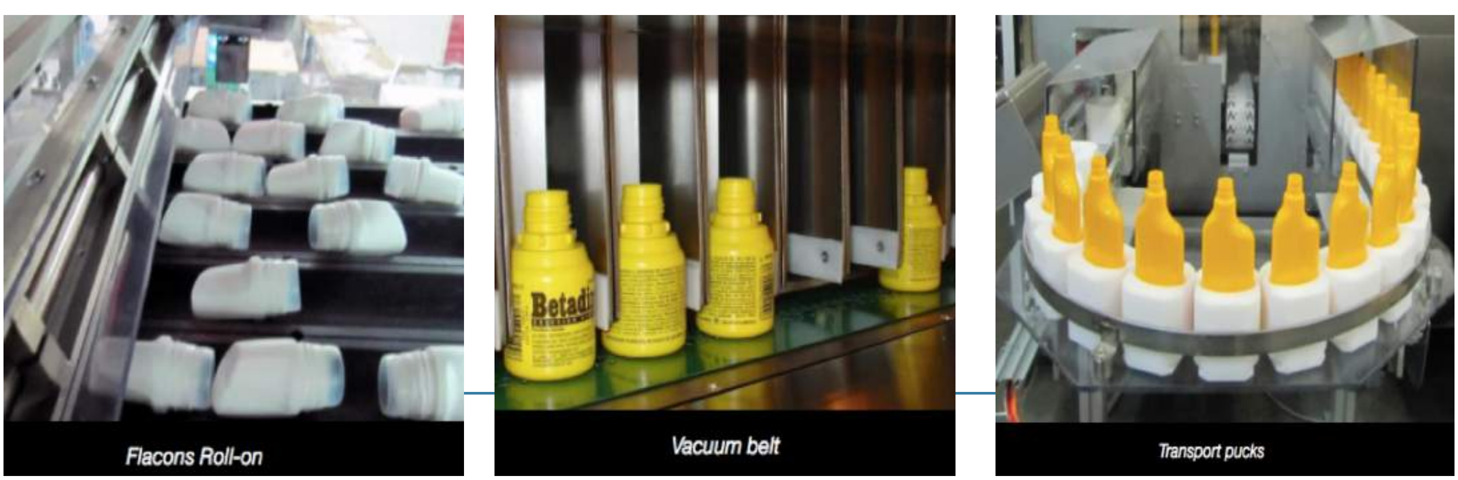


Figure 3 BCM Unscrambler Carrying Bottles Through (A) Vibrating Table (B) Vacuum Belt (C) Transport Pucks [4]

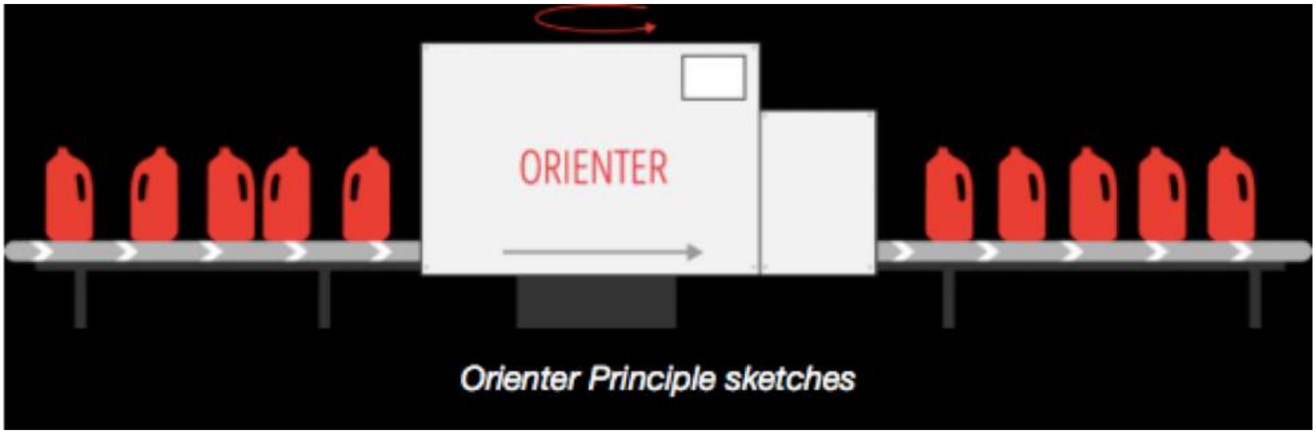


Figure 3 (D) BCM Orienter Orienting the Containers to Align in the Same Direction [5]

4.1.4 Unscrambling and Unjamming Mechanisms

4.1.4.1 Systems Using Centripetal Force

Centrifugal loading bins such as in Figure 1a and Figure 4 are used to unscramble a large number of items, push them towards the bin wall and make them flow through the system without getting jammed. The stationary cone at the centre of the bin in Figure 4 helps the bottles slide towards the wall at all times once the bin starts spinning. Once they are pushed towards the wall via centrifugal force, the bottles fit inside the slots located along the wall and then, moves into an orienting unit.

Figure 4 An Industrial Centrifugal Loading Bin [6]

Figure 5 An Elevator Transporting Bottles Up From A Loading Bin [7]

4.1.4.2 Elevator System

The system in Figure 5 consists of a continuously moving “elevator” (a continuous stream of shelves moving up) which picks up a small portion of bottles from a loading tray and passes them to an entrance to the rest of the system at the top of the elevator. By picking up a fewer number of bottles at a time, this mechanism prevents a large number of bottles from jamming the entrance.

4.1.4.3 Vibrating Mechanisms

Often times vibration or agitation is used on the loading bin to constantly shake items in order to unscramble them and prevent them from getting stuck at any point. An example is the vibrating table with depressed wells in Figure 3A.

4.2 Methods of Identification

4.2.1 Infrared Proximity Sorting

The IR distance sensor in Figure 6 uses a position sensitive detector (PSD) combined with an IR LED to output an analog signal depending on the distance it perceives. A comparator can be used to produce a binary digital signal depending on the desired distance of interest. The range of detection is specified to be 2 - 15 cm which is optimal for the distances and scales that will be encountered. At the current stage, this is the selected sensor for the design, however it is subject to change.

Figure 6 Sharp Gp2Y0A51 Sk0F Analog Distance Sensor From Pololu [8]

4.2.2 Optical Sorting

Optical sorting systems use photo transistors combined with lasers, light sensors, colour sensors, camera, human eye, etc. to visually identify the type or physical state of a bottle.

All three optical sensors mentioned below are possible candidates as replacements for IR proximity sensors depending on the preciseness of the proximity sensors. This is discussed more in detail in Section 5.2 of the report.

Figure 7 (A) Infrared Emitter And Transistor Pair [9], (B) Laser [10], (C) Laser Pointer [11]

4.2.2.1 Phototransistors and Lasers [9,10,11]

The infrared emitter and transistor pair in Figure 7A consists of a LTE-302 infrared emitter which is a 40 degree, 940nm IR emitter rates at 1.5 V 100 mA, and a LTR-301 infrared transistor. The general purpose laser receiver in Figure 7B outputs a high signal when it detects a incoming laser. Either of these two devices can be combined with lasers to detect a transparent or opaque object. The 3.3 V – 5 V red laser pointer in Figure 7C is one such option and is preferred due to its low cost.

4.2.2.2 Light Sensors [12]

The analog and digital light sensor in Figure 8 is based on the GL5528 photo resistor to detect light intensity. The resistance and output voltage of the sensor change depending to the light exposure. This sensor can output both analog and digital signal and a potentiometer can adjust the sensitivity of both. When the light intensity is more than that set by the potentiometer, the terminal outputs HIGH.

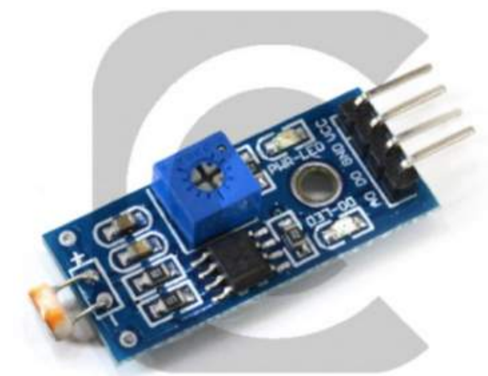


Figure 8 Analog And Digital Light Sensor [12]

4.2.2.3 Colour Sensors [13]

The TCS34725 RGB colour sensor in Figure 9 consists of RGB and clear light sensing elements, an IR blocking filter which allows accurate colour measurements, and a white LED which can turned on or off by a logic level output.

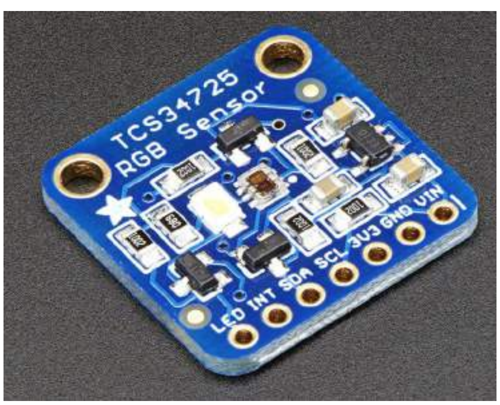


Figure 9 TCS34725 RGB Colour Sensor [13]

Figure 10 Skittles M&M’s Colour Sorting Machine [15]

3.2.2.4 Market Survey: Optical Sorters

Aladdin Optical Sorter is an industrial sorting system by MSS Optical Sorters that uses NIR and colour sensing capabilities to sort by resin and by colour [14]. It can also use a transmission sensor to sort by opacity (transparent PTE bottles vs. opaque HDPE bottles) [14]. Other industrial machines that use optical sorting includes the robotic arm in Figure 2 and the Orienter Machine in Figure 3D, both using camera vision to detect the orientation of the bottles.

Skittles M&M's Colour Sorting Machine is an autonomous Atmel ATmega328-controlled robot (Figure 10) that solely uses RGB colour sensor to detect the colour of candy and to sort them into separate containers [15].

4.3 Methods of Sorting and Delivery

4.3.1 Automated Chutes

Automated chutes that change directions to face a delivery container depending on the identified type of object are often used for delivery. One such example is the feeding tube (Figure 11) used in Skittles M&M's Colour Sorting Machine, previously discussed in Section 3.2.2.4. Once the candy is identified, the feeding tube, attached to a 360-degree servo, starts moving toward the designated cup and when it is within a certain range, the candy is released from the feed wheel and onto the feeding tube. By the time the candy exits the tube, the servo has had time to face towards the designated cup properly. A similar mechanism is used for sorting and delivery in the proposed design.

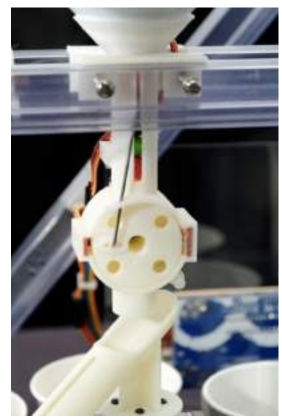


Figure 11 Skittles Sorting Machine Feeding Tube [15]

4.3.2 Conveyor Belts

The design in Figure 12 consists of a conveyor belt that moves a bottle to a certain location and stops to allow a piston to push the bottle to its designated container.

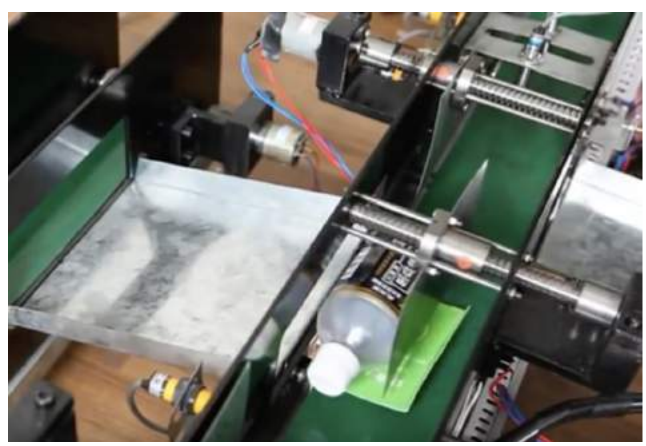


Figure 12 Piston Pushing A Bottle On A Conveyor Belt [16]

4.4 Microcontrollers and Development Environments

4.4.1 Freelance Development Boards and IDEs

Arduino is an easy-to-use electronics platform containing a microprocessor or microcontroller. It is inexpensive and costs less than $50, runs on various operating systems, open source with extensible software and hardware, has an easy-to-use Arduino Software IDE for beginners and supports efficient programming in high-level language such as C. The table below shows a comparison of the different Arduino boards available in market. Among these, the Arduino Mega 2560 is commonly used for complex robotics projects [17].

Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
101	Intel® Curie	3.3 V/ 7-12V	32MHz	6/0	14/4	-	24	196	Regular	-
Gemma	ATtiny85	3.3 V / 4-16 V	8 MHz	1/0	3/2	0.5	0.5	8	Micro	0
LilyPad	ATmega168V ATmega328P	2.7-5.5 V / 2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
LilyPad SimpleSnap	ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	4/0	9/4	1	2	32	-	-
LilyPad USB	ATmega32U4	3.3 V / 3.8-5 V	8 MHz	4/0	9/4	1	2.5	32	Micro	-
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
MKR1000	SAMD21 Cortex-M0+	3.3 V/ 5V	48MHz	7/1	8/4	-	32	256	Micro	1
Pro	ATmega168 ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	0.512 1	1 2	16 32	-	1
Pro Mini	ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	1	2	32	-	1
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1

Zero	ATSAMD21G18	3.3 V / 7-12 V	48 MHz	6/1	14/10	-	32	256	2 Micro	2
Due	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Esplora	ATmega32U4	5 V / 7-12 V	16 MHz	-	-	1	2.5	32	Micro	-
Ethernet	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/4	1	2	32	Regular	-
Leonardo	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1
Yùn	ATmega32U4 AR9331 Linux	5 V	16 MHz 400MHz	12/0	20/7	1	2.5 16MB	32 64MB	Micro	1
Arduino Robot	ATmega32u4	5 V	16 MHz	6/0	20/6	1 KB (ATmega32u4)/ 512 Kbit (I2C)	2.5 KB (ATmega32u4)	32 KB (ATmega32u4) of which 4 KB used by bootloader	1	1
MKRZero	SAMD21 Cortex-M0+ 32bit low power ARM MCU	3.3 V	48 MHz	7 (ADC 8/10/12 bit)/1 (DAC 10 bit)	22/12	No	32 KB	256 KB	1	1

Table 4. Comparison Of Different Arduino Boards Available [17]

Other freelance development boards/IDEs include Raspberry Pi, BeagleBoard/Bone and ARM MBD.

4.4.2 MicroController Units Manufacturer Boards and IDEs

MicroChip Technology and Atmel are companies that offers a wide range of PIC microcontroller families and developer boards. These MCUs support both hardware-level (low-level) and efficient high-level programming, access to peripherals, efficient hardware debugging, The PIC18 family is commonly used for application development and complex robotic projects. A comparison between different PIC18 microcontrollers is provided in Table 5.

The PIC18F4620 microcontroller was used for the design. The high-end architecture of PIC18F4620 is chosen to make use of the large number of I/O pins (36), large memory maps, built-in hardware multipliers for more efficient arithmetic operations and new peripheral features [1]. It also features an extended instruction set to enhance program optimization and built-in C programming optimizing features [1]. Another advantage of using the PIC18

PIC18 Microcontroller Family

Product	Program Memory Type	Program Memory Bytes	Data Memory RAM Bytes	Data Memory EEPROM Bytes	I/O Ports	ADC 10-bit	MSSP	USART	Other	CCP/ PWM	Timers 8/16-bit	Packages	Pins
PIC18F1220	FLASH	4K	256	256	16	7	—	1	6x PMM	1	1/3	DIP, SOIC, SSOP, QFN	18
PIC18F1320	FLASH	8K	256	256	16	7	—	1	6x PMM	1	1/3	DIP, SOIC, SSOP, QFN	18
PIC18F2220	FLASH	4K	512	256	23	10	I^2C/SPI	1	6x PMM	2	1/3	DIP, SOIC	28
PIC18F2320	FLASH	8K	512	256	23	10	I^2C/SPI	1	6x PMM	2	1/3	DIP, SOIC	28
PIC18C242	OTP	16K	512	—	23	5	I^2C/SPI	1	—	2	1/3	DIP, SOIC	28
PIC18C252	OTP	32K	1536	—	23	5	I^2C/SPI	1	—	2	1/3	DIP, SOIC	28
PIC18F242	FLASH	16K	512	256	23	5	I^2C/SPI	1	—	2	1/3	DIP, SOIC, SSOP	28
PIC18F252	FLASH	32K	1536	256	23	5	I^2C/SPI	1	—	2	1/3	DIP, SOIC, SSOP	28
PIC18F258	FLASH	32K	1536	256	22	5	I^2C/SPI	1	CAN 2.0B	1	1/3	DIP, SOIC	28
PIC18F4220	FLASH	4K	512	256	34	13	I^2C/SPI	1	6x PMM	2	1/3	DIP, TQFP, QFN	40/44
PIC18F4320	FLASH	8K	512	256	34	13	I^2C/SPI	1	6x PMM	2	1/3	DIP, TQFP, QFN	40/44
PIC18C442	OTP	16K	512	—	34	8	I^2C/SPI	1	—	2	1/3	DIP, PLCC, TQFP	40/44
PIC18C452	OTP	32K	1536	—	34	8	I^2C/SPI	1	—	2	1/3	DIP, PLCC, TQFP	40/44
PIC18F442	FLASH	16K	512	256	34	8	I^2C/SPI	1	—	2	1/3	DIP, PLCC, TQFP	40/44
PIC18F452	FLASH	32K	1536	256	34	8	I^2C/SPI	1	—	2	1/3	DIP, PLCC, TQFP	40/44
PIC18F458	FLASH	32K	1536	256	33	5	I^2C/SPI	1	CAN 2.0B	1	1/3	DIP, PLCC, TQFP	40/44
PIC18C601	—	ROMless	1536	—	31	8	I^2C/SPI	1	—	2	1/3	PLCC, TQFP	64/68
PIC18C658	OTP	32K	1536	—	52	12	I^2C/SPI	1	CAN 2.0B	2	1/3	PLCC, TQFP	64/68
PIC18F6520	FLASH	32K	2048	1024	52	12	I^2C/SPI	2	—	5	2/3	TQFP	64
PIC18F6620	FLASH	64K	3840	1024	52	12	I^2C/SPI	2	—	5	2/3	TQFP	64
PIC18F6720	FLASH	128K	3840	1024	52	12	I^2C/SPI	2	—	5	2/3	TQFP	64
PIC18C801	—	ROMless	1536	—	42	12	I^2C/SPI	1	—	2	1/3	PLCC, TQFP	80/84
PIC18C858	OTP	32K	1536	—	68	16	I^2C/SPI	1	CAN 2.0B	2	1/3	PLCC, TQFP	80/84
PIC18F8520	FLASH	32K	2048	1024	68	16	I^2C/SPI	2	EMA	5	2/3	TQFP	80
PIC18F8620	FLASH	64K	3840	1024	68	16	I^2C/SPI	2	EMA	5	2/3	TQFP	80
PIC18F8720	FLASH	128K	3840	1024	68	16	I^2C/SPI	2	EMA	5	2/3	TQFP	80

Abbreviation: ADC = Analog-to-Digital Converter; PWM = Pulse Width Modulation; CCP = Capture/Compare/PWM; SPI = Serial Peripheral Interface; I^2C = Inter-Integrated Circuit Bus; USART = Universal Synchronous/Asynchronous Receiver/Transmitter; PMM = Power Managed Mode

Table 5. Comparison Of Different Pic18 Microcontrollers

family is its ability to read and write to/from a hardware implemented stack which allows for the implementation of a Real Time Operating System (RTOS) environment [1].

4.5 Related Research and Theoretical Foundations

4.5.1 Development of Identification System of Cans and Bottles (2015), Sriwijaya University, Indonesia [18]

In this research project, an integrated simulation model of an intelligent sorting system was developed. The system uses machine vision to detect and identify the object and then automatically sort it. The identification system uses pattern recognition through a series of steps such as image acquisition, image pre-processing, feature extraction and identification. The performance of this system is 91.33%, with the success depending on the type and size of the objects. [18]

4.5.2 Development of a Prototype Automated Sorting System for Plastic Recycling (2006), Universiti Kebangsaan Malaysia, Malaysia [19]

This prototype was developed to selectively partition PET plastics from non PET plastics. A specialized hopper was used to allow the objects to enter the testing chamber. In the testing chamber, the identification was accomplished using machine vision for detecting the plastic material through different stages. Once the object is identified, it is ejected to the appropriate container using compressed air. [19]

4.6 Related Patents

4.6.1 Sorting Plastic Bottles for Recycling (US 5314072 A), 1994 [20]

This patented design uses optical sensors accompanied by a truth table for sorting plastic bottles for recycling. First, the transparency of the bottle is checked. If the bottle is transparent, it gets ejected. If it is not, the colour of the bottles is checked. If the bottle is coloured, the bottle gets ejected. Additionally, a detector checks for the presence of chlorine using X-ray fluorescence. Lastly, a sorter device sorts the bottles by diameter. [20]

4.6.2 Container Sorting and Handling System (US 4248389 A), 1981 [21]

This sorting system consists of a sorting station and an infeed conveyor which sequentially and individually transports containers to the sorting station. At the sorting station, a container is rotated and its bar code is checked using an optical scanner. The output signal from the scanner is sent to a computer which compares the signal with pre-stored values. The computer then sends an output signal to an indexing mechanism which transports the container from the sorting station to a one of the multiple conveyors leading to the identified container's designated area. [21]

4.7 Limitations

Cylindrical rotating or centrifugal loading bins, as used in Section 4.1.1, must be manufactured industrially in order to function smoothly and completely. Small defects in the mechanism can lead to jamming in the loading bins. Camera vision used for detection as in Section 4.1.2 requires a high level of software algorithm and electrical hardware complexity. Conveyor belts and transport systems used in Section 4.1.3 and Section 4.3.2 requires high mechanical complexity.

The major limitation of sorting by IR proximity sensors as in Section 4.2.1 is that the object must be within a certain distance for the sensor to accurately and completely detect it. If the object is past the distance, the sensor may or may not detect the object. Moreover, IR sensors have low sensitivity in detecting transparent objects. The limitation of optical sorting, especially using colour sensors and camera vision, as discussed in Section 4.2.2 is that it only detects objects using a restricted/discrete range of colours or patterns. It also requires objects to be in a particular orientation in most cases, unless a highly complex software algorithm is developed to detect objects in any orientation.

The limitations of using Arduino, as in Section 4.4.1, include its inability to support hardware-level (low-level) programming, reparability, limited/ restricted access to peripherals and inefficient hardware debugging. The limitations of using MicroController Unit, such as PIC18-based development platform as discussed in Section 4.4.2 includes a complex programming environment, low speed and low resolution.

5. DESIGN CONCEPT

As mentioned in Section 3.7, the solution was designed with simplicity, safety and modularity in mind, thus focusing on three main mechanisms:

- Loading and Orientation
- Identification
- Sorting and Delivery

The design concept for the orientation mechanism is an actuated V-shaped cradle (Figure 14). The cradle is capable of rocking[2] which agitates the bottles, causing them to fall and directing them lengthwise through the mouth of the cradle. When a bottle exits the cradle, it falls onto a semi-cylindrical, basket-like capture ramp, which guides the bottle into the testing chamber, which can only evaluate one bottle at a time (Figure 15).

Once the bottle enters the testing chamber, it triggers a laser detector. This is the signal for the V-shaped cradle to stop moving and for the upper lid of the testing chamber to close. An infrared break beam sensor checks for the opacity of the bottle. As well, two proximity sensors (one of each lid of the testing chamber) verify whether or not the bottle has a lid. Using the signals from these sensors, the PIC microcontroller determines which of the four possible bottles types, is currently located in the testing chamber.

As soon as this identification is complete, the delivery chute swivels to the appropriate delivery container and the bottle lid of the testing chamber opens (Figure 18). The newly identified bottle now moves into its corresponding bin, via the delivery chute.

Finally, the lower testing chamber lid closes, the upper one opens and the V-shaped cradle once again begins it rocking motion (Figure 13). This cycle is repeated for all of the unsorted bottles contained within the cradle.

[2] The transverse axis is defined as the axis which runs between the sides of the cradle, Figure 13

6. DIVISION OF THE PROBLEM

Within this project, major tasks will be divided into three categories or subsystems, namely, Microcontroller, Circuits and Sensors, and Electromechanical. For the first three weeks, the team worked collectively in order to frame the problem, brainstorm, diverge and converge to a suitable design that will accomplish the given tasks. Each team member then worked independently on their respective subsystem for the following five weeks. However, the team continued to meet at least two times every week. This allowed for updates on progress or setbacks within each individual subsystem to be communicated between members. This facilitated and improved each team member's understanding of the machine as a whole, and allowed for an easier integration and debugging process in the following six weeks.

During the integration phase, as we began testing our machine, it became evident that some electromechanical re-design would be necessary; the whole team came together to aid in and speed up this process to ensure that we would remain on track.

6.1 Processing and Control : Microcontroller

The Microcontroller subsystem lead is primarily responsible for programming the software that is used in the sorting machine. This includes, but is not limited to, keypad and display interface, as well as combinational and sequential logic algorithms. This code can be written in Assembly or ANSI C programming languages. The team lead is responsible for the creating of a micro controller board if budget does not allow for the use of the board provided in the project kit. This member works closely with the Circuits and Sensors subsystem lead in order to ensure that the circuits and code are well integrated.

6.2 Instrumentation and Interfacing : Circuits and Sensors

The Circuits and Sensors subsystem lead is primarily responsible for constructing all digital and analog interfacing elements that is used to connect sensors and actuators to the Microcontroller board. This includes motor or solenoid driver circuits. This member consults with the Electromechanical subsystem lead on the positional placement of the circuits, switches, and sensors on the structural framework of the design. This member is also responsible for complete wiring of the machine and acquiring suitable power supplies.

6.3 Mechanism and Actuation : Electromechanical

The Electromechanical subsystem lead is primarily responsible for the structural fabrication of mounts, frames or platforms required in the machine, and the incorporation of actuators and mechanisms required. This member is also responsible for assigning locations of the sensors and circuit boards within the machine.

7. ELECTROMECHANICAL SUBSYSTEM

7.1 Assessment of the Problem

The autonomous bottle-sorting robot must have these key electromechanical features (Figure 13):

- Loading Chamber
- Orientation Mechanism
- Testing Chamber
- Delivery Bins
- Delivery Mechanism

The loading chamber and delivery bins must hold a maximum of ten bottles, while the testing chamber should only allow for one bottle to enter at a time. The orientation mechanism must position the bottles into a lengthwise orientation that is appropriate for the testing chamber, and the delivery mechanism must accurately transport the identified bottles into the appropriate bin.

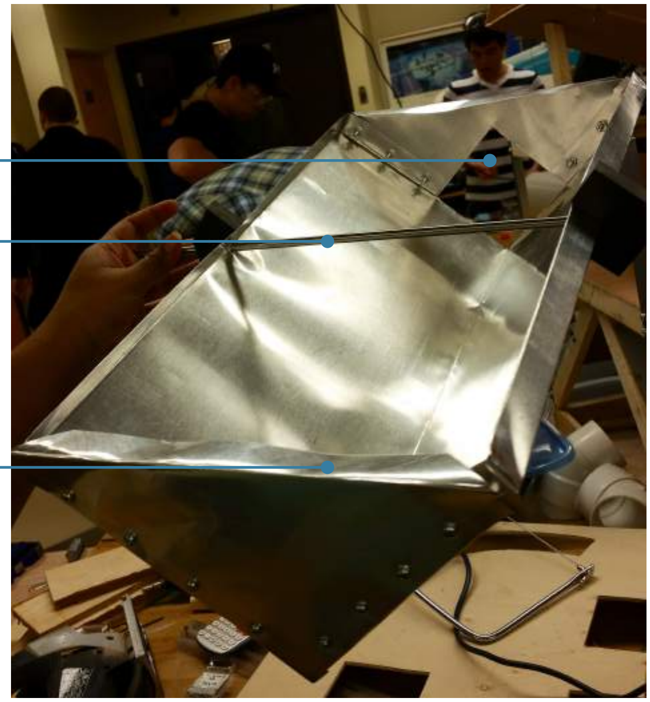
Figure 13. Labelled Image Of Designed Solution

7.2 Solution[3] [4]

7.2.1 Loading Chamber

After a prodigious amount of prototyping, the team decided upon a V-shape for the loading chamber as it facilities the orientation of bottles in the required lengthwise configuration (Figure 14). In addition, actuating this chamber in a vertical rocking motion creates a streamlined flow of bottles. The square opening in the V-shaped cradle is sized to be precisely 2 cm wider than the largest width of the bottles. This ensures that multiple bottles do not fall through the opening at the same time. The cradle also has 2.5 cm wide ledges (folded inwards) across its entire perimeter to prevent bottles from being jostled out during operation.

Figure 14. The V-Shaped Cradle



7.2.2 Orientation Mechanism

The V-shaped cradle is suspended by a metal rod through its centre along the transverse axis. It is located directly above the testing chamber and forward past it. This allows for the correct orientation and delivery of bottles moving into the semi-cylindrical, basket-like capture ramp. The forward placement of the cradle angles its opening such that, at the extreme position of the cradle tilted towards the testing chamber, the opening is just slightly above the capture ramp (angled at 30°) (Figure 15). This flips the bottle along its lengthwise orientation and

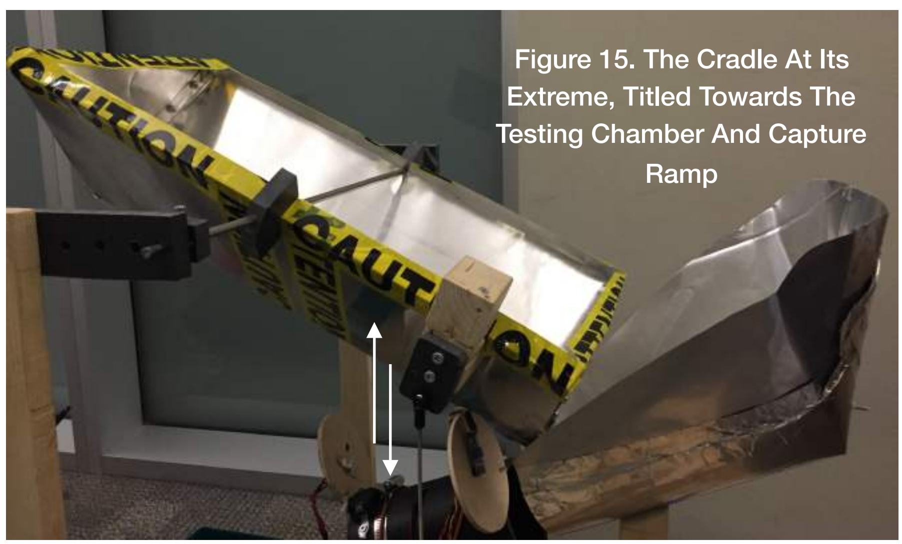


Figure 15. The Cradle At Its Extreme, Titled Towards The Testing Chamber And Capture Ramp

[3] Materials used for all mechanisms described are listed in Table 6 below

[4] See Appendix C for all relevant calculations

provides it with enough momentum to easily slide into the testing chamber.

The cradle itself is actuated using a piston-like mechanism that is powered by a DC Gearbox Motor. The shaft of the motor is coupled to a plastic arm using a set screw. A threaded rod is attached to the opposite end of this plastic arm using a spherical rod end. The upper end of the threaded rod is connected to another piece of plastic on the side of the V-shaped cradle, also using a spherical rod end. The spherical rod ends facilitate the 360°-motion of the threaded rod. This gives the piston-like mechanism the freedom to move with reduced friction, along a circular path. (Figure 16)

<— **Figure 16. The Piston-Like Mechanism That Actuates The Cradle**

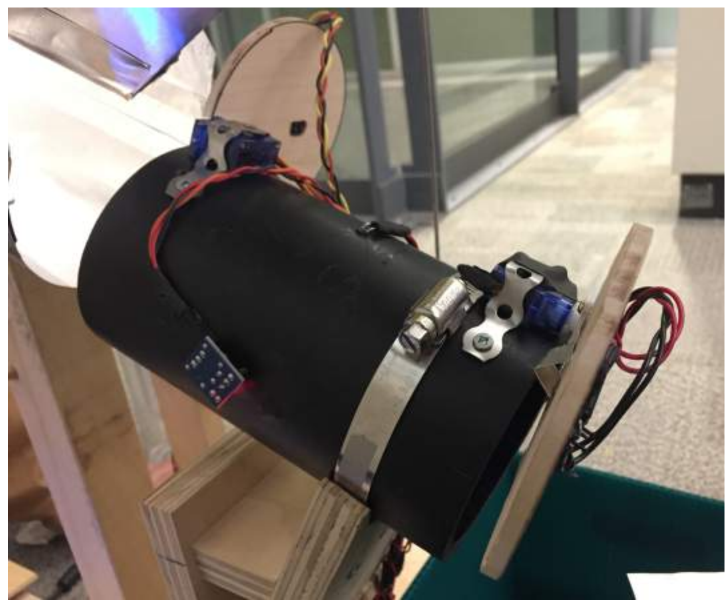


Figure 17. The Testing Chamber

7.2.3 Testing Chamber

The testing chamber is a 7.5 cm diameter cylindrical compartment that is approximately 16 cm long (Figure 17). It has circular lids, controlled by 180° servo motors. The lids rotate open such that they are vertically above the cylindrical chamber. The testing chamber is placed at a 30° angle above the horizontal to allow for gravity-aided motion of bottles into the chamber. The chamber itself, and its lids also have precisely placed and sized holes, where sensors are embedded.

It is vital that the chamber is a stable and modular entity as it is a key element in the identification of bottle types; stability ensures that the bottles are not jostled overly within the chamber, allowing for accurate sensor readings; modularity ensures that the chamber is easily replaceable in case of worn-out or damages sensors or motors.

7.2.4 Delivery System

The delivery system is a semi-cylindrical ramp that channels bottles from the testing chamber to the appropriate container. The ramp is located below the testing chamber, but on the same central axis. It is angled at 45° such that it aims right into the delivery bins. As with the testing chamber, this allows for the gravity-aided motion of bottles, and also for accurate delivery. The ramp is controlled by a 360° continuous servo motor.

If the delivery ramp is considered the origin, then the four delivery bins are located in the four quadrants around it (Figure 18). The bins are held in place by a frame.

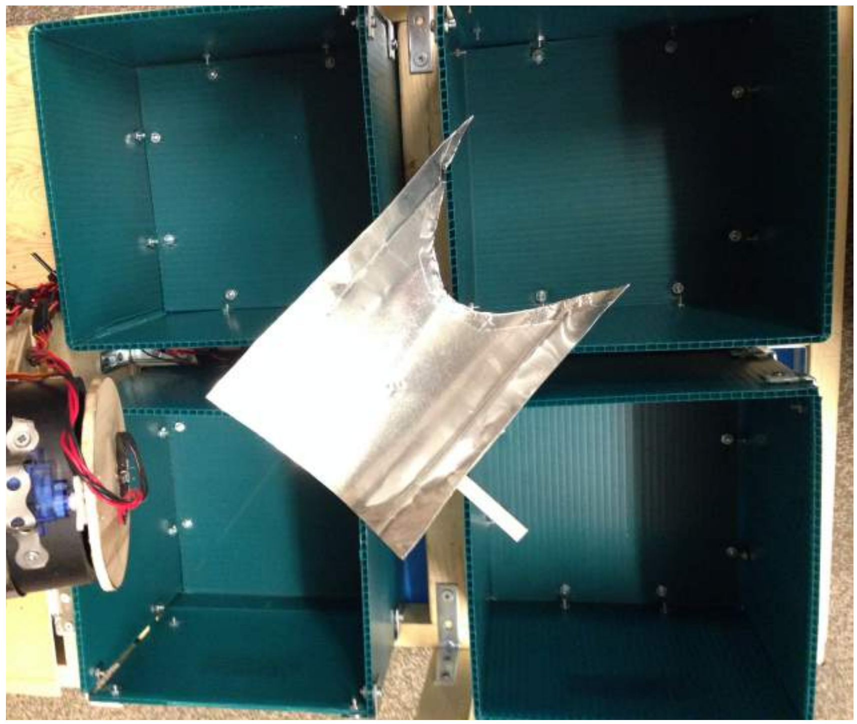


Figure 18. The Delivery Chute And Delivery Containers

Device Part	Materials Used
V-Shaped Cradle	✓ 28 gauge aluminum sheet metal ✓ 0.25” diameter solid metal rod ✓ Self-fabricated shaft collars (using plastic and set screws)
Orientation Mechanism	✓ 6V, 7.9 kg cm Torque DC Metal Gearbox Motor ✓ Threaded rod ✓ Spherical Rod Ends w/ appropriate screws ✓ Self-fabricated piston arm (using plastic)
Capture Ramp	✓ 28 gauge aluminum sheet metal ✓ Spruce beams

Device Part	Materials Used
Testing Chamber	✓ 3” pipe clamp ✓ 3” (diameter) x 0.5’ (length) ABS pipe ✓ 0.5” plywood ✓ 0.25” plywood ✓ 2 x 6V, 1.3kg cm torque FS90 180° Servo Motor
Delivery Chute	✓ Spruce beams ✓ 28 gauge aluminum sheet metal ✓ 6V, 1.5kg cm torque FS90R Continuous Servo Motor
Delivery Containers	✓ Corrugated plastic sheet
Frame	✓ Spruce beams ✓ 0.25’ (thickness) Plywood ✓ Corrugated Plastic Sheet
Miscellaneous	✓ Screws (various lengths and sizes) ✓ Bolts (various lengths and sizes) ✓ Washers (various sizes) ✓ Corner braces (various sizes) ✓ Aluminum tape ✓ Epoxy ✓ Hinges

Table 6. Materials Used To Construct Each Mechanical Component

7.3 Suggestions for Improvement of the Subsystem

Device Part	Limits and Problems	Potential Solutions and Suggested Improvement
V-Shaped Cradle	Very rarely jamming occurs in this chamber, due to space constraints and the positioning of the rod the suspends it The clogging issue was reduced by making the cradle bigger (it is currently twice its initial size)	This can be prevented by: • Increasing the length and width of the cradle • Using a different method to suspend the chamber, such as two short vertical rods from either attached to either end of the cradle • Vibration pads along the cradle to provide further agitation

Device Part	Limits and Problems	Potential Solutions and Suggested Improvement
	Sudden jarring of bottles at the extreme positions of the cradle, causes bottles to sometimes fly out of the cradle	This can be prevented by: • Creating a fully enclosed cradle with a door though with bottles can be deposited • Increasing the maximum tilt of the cradle, and slowing the rocking mechanism
Orientation Mechanism	This mechanism is inconsistent. The motor seems to weaken and strengthen sporadically.	This can be improved by: • Using a stronger torque motor • Using a different method to create the rocking motion required, such as a rack and pinion
Capture Ramp	Sometimes bottles fly out of the cradle and over the capture ramp To mitigate this, we added walls to our capture ramp, but there was still cases of flyaway bottles	This can be prevented by: • Slowing the rocking motion of the cradle • Changing the shape of the ramp to add higher walls, with inward ledges (similar to the cradle)
	Jamming occurs at the lower end of this ramp, where it meets the testing chamber	This can be prevented by: • Changing the shape of the ramp so that it is semi-conical, such that the lowest diameter of the cone exactly match the diameter of the testing chamber • Eliminating the gap between the capture ramp and the testing chamber
Testing Chamber	Bottles crash into the lower lid of the chamber, speeding up the wear-and-tear of the actuators over time	This can be prevented by: • Adding some form of speed bump at the entrance of or slightly inside the chamber • Inserting a removable obstruction right before the bottom lid, which acts as a shock absorber
	Bottles sometimes land at slightly angled positions inside the chamber, creating inaccuracies in the identification process	This can be prevented by: • Reducing the diameter to the chamber such that it closely fits the maximum diameter of the largest bottle

Device Part	Limits and Problems	Potential Solutions and Suggested Improvement
	The lids of the chambers sometimes interfere with the cradle's motion This issue was mostly overcome by reducing the height of the arm that holds up the testing chamber; however, the issue still sometimes occurs.	This can be prevented by: • Increasing the clearance between the cradle and the testing chamber • Using rectangular or polygonal partial lids, that leave the entrance and exit free of obstruction with only a 90° rotation to the side of the chamber
Delivery Chute	This mechanism is inconsistent, as bottles sometimes don't have enough energy to move down the ramp.	This can be improved by: • Increasing the slope of the delivery chute • Adding some form of actuation on the ramp to facilitate the motion of bottles, such as vibration pads or solenoids

Table 7. Limits And Problems Of The Currents Design And Potential Improvements

8. CIRCUITRY AND SENSORS

8.1 Assessment of the Problem

The circuitry onboard the machine is responsible for connecting the sensors and actuators to the input and output pins of the PIC microcontroller. The circuits and sensors can be separated into the following roles:

- Bottle detection
- Bottle identification
- Motor control
- Power control

A bottle detection system is required since the design of the testing chamber involves opening and closing lids; the system must be able to know when is the appropriate time to close the lids and identify the bottle. Identification of the bottle into four different types depends on two physical traits: Yop vs Eska and Capped vs Uncapped. These traits need to be measured in a bottle identification system. In addition, the movement of the DC motor and continuous servo motor has to be controlled, so a circuit responsible for signalling the start and stop of these motors is required. In particular, the continuous servo motor accumulates offsets which decreases the accuracy of the delivery stage.Finally, the correct currents and voltages need to be supplied to each component, which have different requirements. In case of an emergency, an emergency stop switch is required to be implemented. The power control system is responsible for these features.

8.2 Solution

8.2.1 Bottle Detection

A laser diode and a laser detector module are placed facing each other on the walls on the testing chamber. This acts as a break beam system -- when there is no bottle in the chamber, the laser detector will detect the full intensity of the laser, and when a bottle falls inside of the chamber, the light from the laser is either blocked (in the

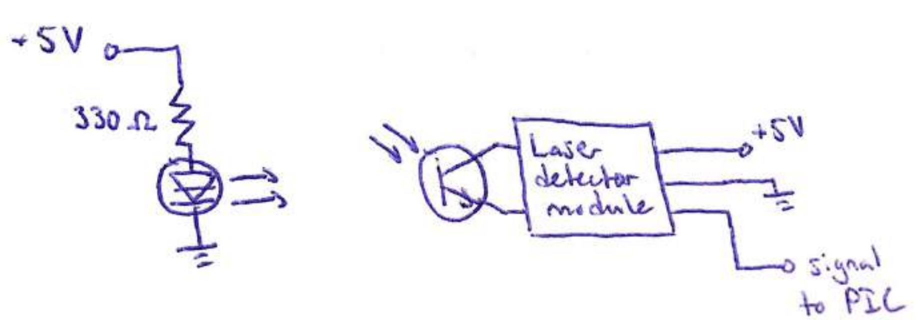


Figure 19. Laser Break Beam System

case of Yop) or scattered (in the case of Eska). Thus, when there is no bottle inside the chamber, the output of this system will be a high, and when there is a bottle, this system will output a low to the PIC.

A module was used in this system for two main reasons:

1. Convenient output/input values; 5V input and digital signal output
2. Shape; the size and shape facilitates mechanical integration to the testing chamber

The triggering of this system controls both the closing/opening of the top lid as well as the starting/stopping of the DC motor which actuates the rocking motion of the cradle.

8.2.2 Bottle Identification

Since the identification of two binary physical traits -- Yop vs Eska and Cap vs Uncapped -- are able to fully classify the bottle, two different identification systems will be used; one for each characteristic.

8.2.2.1 Differentiating between Yop and Eska

One easily detectable characteristic to determine the Yop vs Eska trait is the opacity of the bottles. If the bottle is opaque, then it can be classified as Yop. Contrarily, if the bottle is transparent, then it can be classified as Eska. In order to measure the opacity, an IR break beam system is used. This system consists of an LTE-5208a IR emitter and an LTR-3208e IR phototransistor receiver. The IR beam from the emitter is able to penetrate Eska bottles and be detected by the receiver, while the Yop blocks the IR and it does not get detected by the receiver. In the IR break beam system used by this machine, the output is high if the IR is blocked (Yop) and the output is low if the IR is not blocked (Eska).

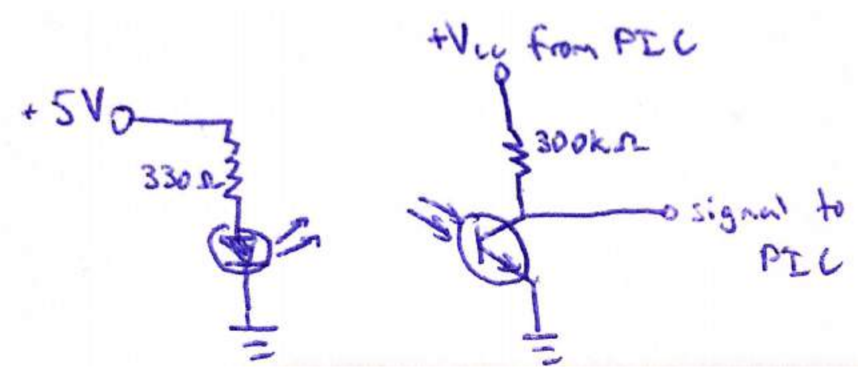


Figure 20. IR Break Beam System

The Yop vs Eska classification is done before the detection of the cap. This reason for this is explained in the algorithm.

8.2.2.1 Detecting the Presence of a Cap

Like the classification of the bottle type, the presence of the cap is also detected by IR sensors. In this case, the sensor system is composed of IR reflection sensors instead of a break beam sensor. The TCRT5000 consists of an IR emitter and a phototransistor side-by-side. When an opaque object is in close proximity with the sensor, the IR from the emitter will bounce off and reflect back into the phototransistor.

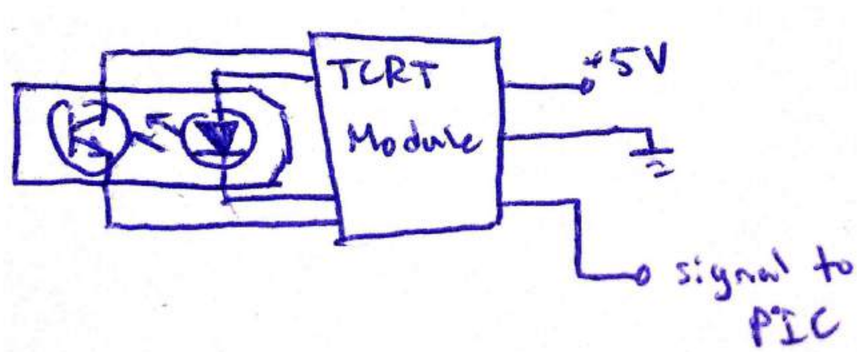


Figure 21. TRCT5000 Module Ir Reflection Sensor

There is one TCRT5000 on each lid in order to cover both possible positions of the cap. When a cap or the bottom of a Yop bottle is against an IR reflection sensor, the phototransistor will be activated. For the bottom lid, this means that the system will output an analogue low to the PIC. For the top lid, a TCRT5000 module is used and will output a digital high if the phototransistor is activated.

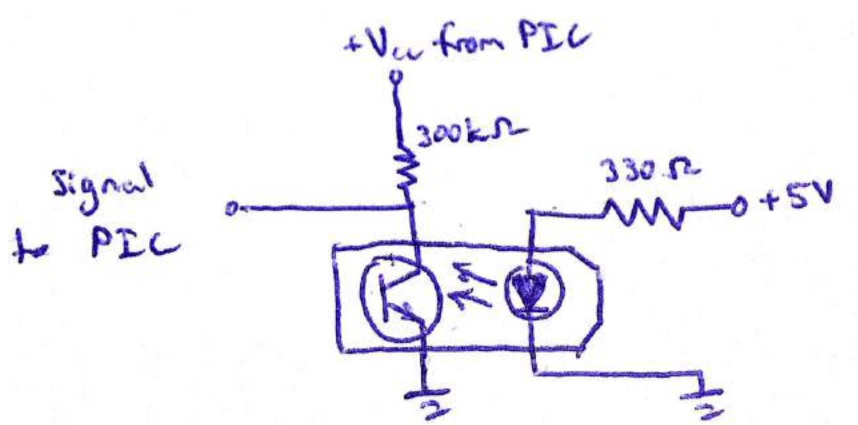


Figure 22. TCRT IR Reflection Sensor

A TCRT module is used for the top lid because of the different heights between Eska and Yop bottles. If an analogue signal is received at the top lid, then one would expect a large range of values for Eska and Yop bottles due to the heights and differences in proximity. However, the digital output of the module ensures that a consistent high signal is outputted regardless of Yop or Eska. It is acceptable for the bottom lid to contain a TCRT5000 without a module since the analogue signal is expected to be fairly consistent since gravity pulls the bottles down the chamber and ensures little variation in proximity at the bottom of the chamber.

8.2.3 Motor Control

There are three types of motors being used: 180 Servo motors, DC motors, continuous servo motors. The 180 servo motors are controlled by PWM directly from the PIC. The DC motor and continuous servo motor require slightly more complex stop conditions.

8.2.3.1 DC Motor Control

The important feature of the rocking mechanism is the precise cycle and speed during which the motor rotates. In order to control the cycle, the DC motor is turned on and off by one input to H-bridge being toggled between high and low. The other input is kept at a constant high. When the control input is a high, the motor will be stopped since there is no electromotive force across the motor. When the control input is low, the voltage across the motor will drive the motor.

In order to control the speed of the motor, PWM is used. The PWM is connected to the enable signal to the H-

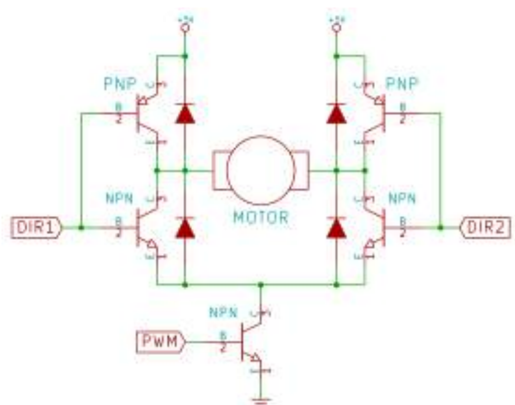


Figure 23. An H-Bridge Circuit With 2 In Signals And 1 Enable [22]

bridge. This allows the average voltage applied across the motor to be changed, thus controlling the speed.

8.2.3.2 Continuous Servo Motor Control

Movement of the continuous servo which controls the delivery chute is composed of two main phases: the selection phase and the return phase. During the selection phase, a PWM signal from the PIC over a determined period of time will move the delivery chute from the default position to the appropriate position. During the return phase the delivery chute is rotated back to its default position. In order to combat any offset accumulation, an IR reflection sensor is used to help the delivery chute to return.

The chute is instructed to rotate until the IR sensor is triggered. However, there is a delay between when the sensor is triggered and when the motor is stopped. At the utilized PWM, the delay was measured to be 45 degrees. Thus, an appendage was added to the motor to trigger the IR reflection sensor 45 degrees before the default position, canceling out the delay.

Figure 24. Delivery Chute With Breakbeam Sensor

8.2.4 Power Control

A 12 V, 36 W power supply is used for the machine. There is a factor of safety of 1.27 for the worst case. Below is a table with worst-case power consumptions.

See Appendix C for Calculations

Component	Qty	Power Cons. per unit (W)	Comments
SG90 Servo	2	3.25	assuming stall
FS90R Servo	1	2.75	assuming stall
TCRT5000	3	0.2	
LTR-3208e	1	0.1	
LTE-5208a	1	0.15	
Laser Diode	1	0.0758	calculated
Laser Detector	1	0.1	assuming same as LTR-3208e
DC Gear Motor	1	18	assuming stall
Total		**28.3**	

Note that the most demanding components -- the motors -- do not operate simultaneously. This was designed to alleviate stress on power demands.

8.2.4.1 Supplying the Correct Voltage

A lot of components take 5 V as their input voltage, rather than the 12 V provided by the power supply. Three voltage regulators are used to obtain 5 V. Using a single voltage regulator would be dangerous due to high current draw and overheating. The motor driver and the PIC both receive the full 12 V as required.

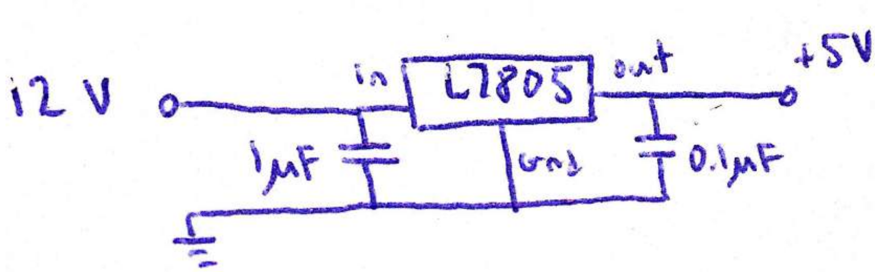


Figure 25. L7085 (5V) Voltage Regulator Setup

8.2.4.2 Emergency Stop

An button placed in series on the 12 V input from the power supply but before the main circuit will serve as the emergency stop. This set up will cut off all power immediately to the circuit and moving components and the PIC once it is pressed.

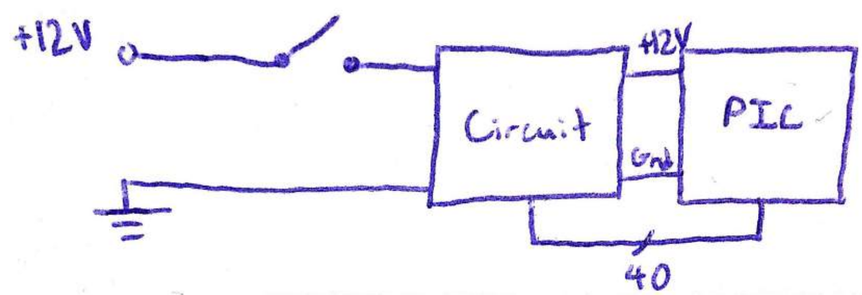


Figure 26. Emergency Stop Switch

8.3 Suggestions for Improvement of the Subsystem

8.3.1 Bottle Detection

Is it possible for an Eska bottle to fall inside of the testing chamber in a position where the laser detector does not detect the presence of a bottle. This likely happens because slight deformations in the bottle may result in less scattering of the laser beam, so the laser detector still receives a significant intensity of the laser. A possible improvement for the detection system could be the inclusion of a mechanical switch instead of a laser break beam. The advantage of a mechanical switch is the lower dependence of position randomness and potentially easier set up. The laser break beam was difficult to set up due to the precision required between the beam and the detector.

8.3.2 Bottle Identification

A Yop bottle can be misidentified as an Eska bottle if the IR receiver is falsely activated. This can happen due to interference between the three IR sensor systems inside of the testing chamber. Rarely, the sensor responsible for Yop vs Eska classification can be activated by the two IR emitters on the lids of the chamber. In addition to mechanical adjustments, the substitution of the IR break beam for a colour sensor as described in 4.2.2.3 for the Yop vs Eska classification can alleviate this issue.

8.3.3 Motor Control

The main complications which arise form timed PWM periods of the DC motor and continuous servo motors are the result of offsets which accumulate. These are also very difficult to fix by changing the code, as the degree of offset may change from day-to-day and even between consecutive trials. Although the inclusion of an IR sensor for the continuous servo helps the offset issue for the servo, it adds additional complexity to the design. The substitution of the continuous servo and DC motor for steppers motors would be an improvement. Stepper motors are capable of 360 degree rotation in discrete and fixed steps. For the 360 degree rotation of the DC motor and the multiple of 90 degree rotations required from the continuous servo, a stepper motor would be a simple resolution of the offset issue.

8.3.4 Power Control

One concern of the current design is the large number of components which require 5 V input despite the 12 V power supply. Voltage regulators dissipate a lot of heat from power losses, which would not happen if a more suitable power supply and component combination were used. A separate power supply which outputs 5 volts could be used in the machine. In addition, using a DC motor rated for 12 V and higher torque is valid possibility which improves on the current design.

9. MICROCONTROLLER

A PIC DevBugger Rev 4.0 development board by Personal Mechatronics Lab with a PIC18F4620 microcontroller was used to program the machine. The technical specifications of the microcontroller can be found in Section 4.4.2. The features of the development board is provided below.

2 Features

- Open, modular, and simple design for learning purposes
- Supports 18-, 28-, and 40-pin packages for PIC16 and 18 families
- In-circuit PICkit™ 3 compatible High Voltage Programmer (works with MPLABX® IDE)
- Dual power source, USB or DC power supply input (from 7.5VDC to 17VDC)
- Dual system voltage support (5V and 3.3V system VCC)
- Debugging Module with 32 indicator LEDs and signal-emulation switches
- Keypad encoder chip is a full featured PIC, whose firmware can be modified for extra memory, parallel processing, and/or I/O pin extension for the primary MCU
- Real Time Clock peripheral with 32.768khz crystal and battery socket (RTC chip included)
- On-board socket for Hitachi HD44780 compatible LCD with contrast and backlight controls, support size up to 4x20 characters
- On-board 4x4 matrix keypad socket and PIC based encoder
- 40-pin I/O bus with ribbon cable connector
- Interchangeable oscillator clock (10MHz crystal included), with dedicated oscillator socket for 18-pin PIC.
- I^2C bus expansion socket
- On-board adjustable ADC voltage reference
- Arduino Nano board interface, with individually selectable pin pass-through to PIC.
- Arduino shield connector, with pin pass-through from both PIC and Arduino.

Figure 27. Features Of Pic Devbugger Development Board

MPLAB X IDE v3.55 by Microchip Technology was used to write the code in C language. MPLAB XC8 Compiler v1.41 by Microchip Technology was used to compile the code for the machine.

9.1 Assessment of the Problem

Optimized code must be developed for major components of the design:

- Loading bin DC motor actuation
- Testing chamber top and bottom lid servo actuation
- Identification algorithm using digital and analog IR sensors
- Delivery chute continuous servo rotation

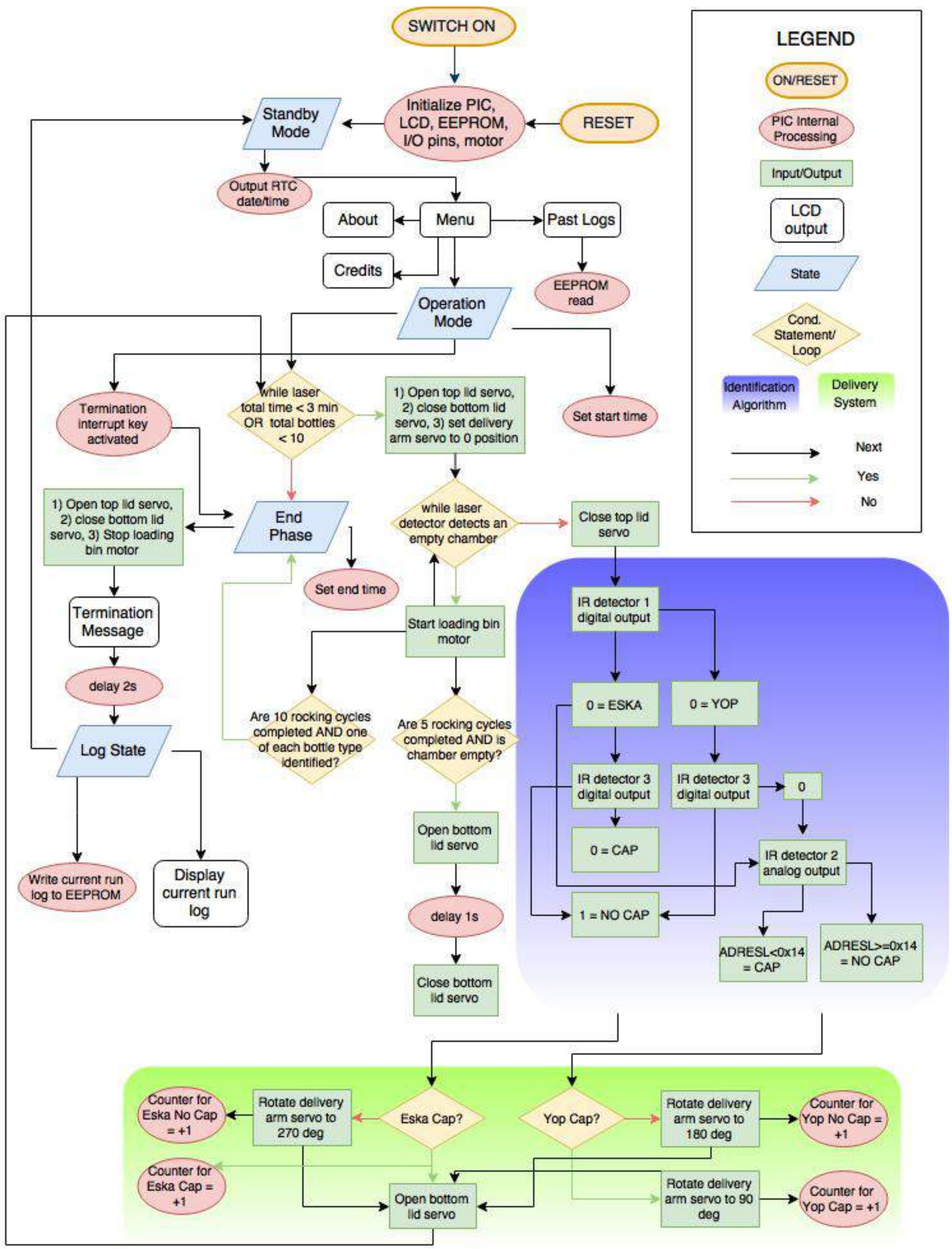


Figure 28. Flowchart Outlining The Processes Executed By Pic

- LCD user interface (user prompts, permanent logs, standby date/time)

The DC motor controlling the loading bin must be programmed using Pulse Width Modulation (PWM), which controls the power delivered to the motor and hence, its speed. However, the PWM must be adjusted in order to achieve full functionality of the loading bin, which should ideally be able to dispense a bottle per rocking cycle. The continuous servo attached to the delivery chute also requires PWM, which must be adjusted to rotate the chute to different degrees. Code must be written for digital input/output and analog input from different pins on the microcontroller. All actuators and sensors are controlled through the input/output pins on the PIC.

Other PIC programming includes handling keypad interrupts, selection of input/output pins, PIC18F5620 bit configuration and, LCD and Inter-Integrated Circuit (I2C) initialization.

9.2 Solution

9.2.1 Overview

Figure 28 shows a flowchart outlining the sequence of tasks executed by the PIC microcontroller to automate the machine. Refer to Appendix D for the complete C code used to program the machine.

9.2.2 Input/Output Pin Assignments

Funtion	I/O	A/D	Pins
Testing chamber laser detector signal	I	D	RC0
Testing chamber bottom lid IR sensor (TCRT5000)	I	A	RA2
Testing chamber body IR detector	I	D	RC6
Testing chamber top lid IR sensor (TCRT5000 module)	I	D	RA4
Loading bin DC motor actuation	O	D	RE0, RB0, RB2
Interface with 4x4 keypad	I	D	RB1, 4:7
Testing chamber bottom lid servo actuation (FS90)	O	D	RC5
Testing chamber top lid servo actuation (FS90)	O	D	RA5
Delivery chute continuous servo actuation (FS90R)	O	D	RC7
Delivery chute IR sensor signal (TCRT5000L)	I	A	RA3
Interface with Real Time Clock (RTC)	I	D	RC3:4

Funtion	I/O	A/D	Pins
Interface with LCD (HD44780)	O	D	RD2:7
Interface with main oscillator	I	D	RA6:7

Table 9. I/O Pin Assignments

9.2.3 PIC18F4620 Bit Configuration

The configuration settings for the program is provided below.

```
// CONFIG1H
#pragma config OSC = HS         // Oscillator Selection bits (HS oscillator)
#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF       // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)
// CONFIG2L
#pragma config PWRT = OFF       // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = SBORDIS  // Brown-out Reset Enable bits  (SBOREN is disabled)
#pragma config BORV = 3         // Brown Out Reset Voltage bits (Minimum setting)
// CONFIG2H
#pragma config WDT = OFF        // Watchdog Timer Enable bit (WDT disabled)
#pragma config WDTPS = 32768    // Watchdog Timer Postscale Select bits (1:32768)
// CONFIG3H
#pragma config CCP2MX = PORTC   // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON      // PORTB A/D Enable bit (PORTB<4:0> pins as analog input on Reset)
#pragma config LPT1OSC = OFF    // Low-Power Timer1 Oscillator Enable bit (Timer1 for higher power operation)
#pragma config MCLRE = ON       // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)
// CONFIG4L
#pragma config STVREN = ON      // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = OFF        // Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
#pragma config XINST = OFF      // Extended Instruction Set Enable bit (Legacy mode)
// CONFIG5L
#pragma config CP0 = OFF        // Code Protection bit (Block 0 (000800-003FFFh) not code-protected)
#pragma config CP1 = OFF        // Code Protection bit (Block 1 (004000-007FFFh) not code-protected)
#pragma config CP2 = OFF        // Code Protection bit (Block 2 (008000-00BFFFh) not code-protected)
#pragma config CP3 = OFF        // Code Protection bit (Block 3 (00C000-00FFFFh) not code-protected)
// CONFIG5H
```

```
#pragma config CPB = OFF        // Boot Block Code Protection bit
#pragma config CPD = OFF        // Data EEPROM Code Protection bit (Data EEPROM not code-protected)
// CONFIG6L
#pragma config WRT0 = OFF       // Write Protection bit (Block 0 (000800-003FFFh) not write-protected)
#pragma config WRT1 = OFF       // Write Protection bit (Block 1 (004000-007FFFh) not write-protected)
#pragma config WRT2 = OFF       // Write Protection bit (Block 2 (008000-00BFFFh) not write-protected)
#pragma config WRT3 = OFF       // Write Protection bit (Block 3 (00C000-00FFFFh) not write-protected)
// CONFIG6H
#pragma config WRTC = OFF       // Configuration Register Write Protection bit
#pragma config WRTB = OFF       // Boot Block Write Protection bit
#pragma config WRTD = OFF       // Data EEPROM Write Protection bit (Data EEPROM not write-protected)
// CONFIG7L
#pragma config EBTR0 = OFF      // Table Read Protection bit
#pragma config EBTR1 = OFF      // Table Read Protection bit
#pragma config EBTR2 = OFF      // Table Read Protection bit
#pragma config EBTR3 = OFF      // Table Read Protection bit
// CONFIG7H
#pragma config EBTRB = OFF      // Boot Block Table Read Protection bit
```

9.2.4 LCD and I2C Initialization

Refer to Appendix D-2 and D-3 for the complete code for LCD and I2C initialization.

9.2.5 Loading Bin DC Motor Actuation

The loading bin is tilted to its lowest point towards the capture ramp prior to the operation. Once the operation is started, the loading bin is rocked through one cycle using PWM. When the loading bin returns back to its original position, the motor is stopped and a delay of 3 seconds is executed before the next rocking cycle begins. The stop condition along with the delay allows a sliding bottle to completely exit from the loading bin opening onto the capture ramp.

Table 10 shows the signals needed for three digital input pins-enable, input1 and input 2, to control the motor through a motor driver (L298D).

Motor Status	Enable Pin (RE0)	Input 1 Pin (RB2)	Input 2 Pin (RB0)
Clockwise	1	1	0
Anti-Clockwise	1	0	1

Motor Status	Enable Pin (RE0)	Input 1 Pin (RB2)	Input 2 Pin (RB0)
Stops	0	1	1
Stops	0	0	0

Table 10. Control Signals And Motor Status

The loading bin rocks back and forth while a laser detector in the testing chamber sends a digital input of 1 to PIC, thus detecting an empty chamber. Once there is a bottle detected inside the testing chamber (digital output of 0 from the laser detector), the rocking motion stops and the next part of the code is executed. The code snippet for actuating the loading bin motor is provided below.

```
RB2 = 1;        //RB2 is the output pin for DC motor input 1 pin
RB0 = 1;        //RB0 is the output pin for DC motor input 2 pin
...
while (RC0 && curr_state==OPERATION){
        //wait here until laser detector (RC0) detects a bottle in the
        //testing chamber; 1 = no bottle, 0 = bottle

        RB0 = 0;        // motor start
        for (unsigned int i = 0; i<31; i++){    //PWM for motor
            PORTEbits.RE0 = 1;          //RE0 is the output pin for DC motor enable pin
            __delay_us(18300);
            PORTEbits.RE0 = 0;
            __delay_us(1700);
        }
        RB0 = 1;            //stop motor for 3s after every cycle
        __delay_ms(3000);
         ...
}
```

9.2.6 Testing Chamber Lid Servo Actuation

The testing chamber lids are controlled by two 180 degree servo motors, which are signalled by digital output pins RA5 for top lid servo and RC5 for bottom lid servo. A digital input of 1 activates and 0 deactivates the servos. The servos reach certain degrees to open and close the lids using PWM. PWM used to set a certain degree was dependent on the servo attachments on the testing chamber. The code was written once the servos were epoxied to the chamber. Hence, the degree used to open and close the lids would need to change if the servos are displaced.

Prior to the loading bin actuation, the top servo opens the top lid to accept a bottle and the bottom servo closes the bottom lid to catch the bottle in the testing chamber. The loading bin is then actuated as per Section 9.2.4 until the laser detector in the testing chamber detects a bottle in the chamber. At this point, the top servo closes the top lid and the identification algorithm begins, as will be discussed in Section 9.2.6. After the identification stage, the delivery chute rotates to the corresponding container for the identified bottle, as will be discussed in Section 9.2.7, and the bottom servo opens the bottom lid to allow the bottle to be delivered to its container. After the operation is terminated, the top and bottom servos return the lids to their default position: open top and close bottom.

During the loading bin actuation, the bottom servo opens the bottom lid after 5 rocking cycles with no detection of bottle made by the laser detector. This is a foolproof mechanism used to allow a bottle, if any, in the testing chamber that the laser detector may have failed to detect to pass through the chamber and into a delivery container. Even though the bottle may not be delivered to the correct container, this mechanism helps prevent any jamming from occurring in the capture ramp.

Below is a code snippet of how PWM is used to open and close the top. A similar code was written for the bottom lid actuation.

```
void opentoplid(void)
{
   unsigned int i;
      for(i=0;i<50;i++)              //PWM for top 180 deg servo
      {
        PORTAbits.RA5 = 1;           //RA5 is output pin for top lid of testing chamber
        __delay_us(1200);
        PORTAbits.RA5 = 0;
        __delay_us(18800);
      }
}

void closetoplid(void)
{
   unsigned int i;                   //PWM for top 180 deg servo
```

```
        for(i=0;i<50;i++)
        {
          PORTAbits.RA5 = 1;
          __delay_us(2700);
          PORTAbits.RA5 = 0;
          __delay_us(17300);
        }
}
```

9.2.7 Identification Algorithm Using IR Sensors

Three Infrared (IR) sensors were used for the identification process: two digital output and one analog output. Once both the top and bottom lids of the testing chamber are in closed position, the identification process begins.

First, the IR detector on the body of the testing chamber sends a digital input to the PIC. An input of 1 means the bottle is opaque (Yop), while an input of 0 means the bottle is transparent (Eska). Next, digital input from the TCRT5000 IR module on the top lid and analog input from the TCRT5000 IR sensor on the bottom lid are sent to the PIC. The bottle is identified as Yop with cap if all three conditions are met: 1) the input from IR detector is 1, 2) input from TCRT5000 module is 0 and, 3) the analog input from TCRT5000 sensor has an ADRESL of less than 0x14. If the first condition is met but both or one of the last two conditions are not met, the bottle is identified as Yop with no cap. The bottle is identified as Eska with cap if the input from the IR detector is 0 and, one or both of the two conditions are met: the input from TCRT5000 module is 0 or the analog input from TCRT5000 sensor has an ADRESL of less than 0x14. Otherwise, the bottle is identified as Eska with no cap.

The code snippet for the identification algorithm is provided below.

```
    readADC(2); //reads analog input from RA2 (IR TCRT5000) on bottom lid of testing chamber

        if (RC6){ //RC6 is IR detector on testing chamber body. input 1 for opaque, 0 for clear
            brand = YOP;
        }
        else if (!RC6){
            brand = ESKA;
        }

        if (brand == YOP && RA4 == 0 && (ADRESL < 0x14)){
            //RA4 is IR TCRT5000 module on top lid of testing chamber
```

```
    type = CAP;
}
else if (brand == ESKA && (RA4 == 0 || (ADRESL < 0x14))){
    type = CAP;
}
else{
    type = NOCAP;
}
```

9.2.8 Delivery Chute Servo Rotation

The delivery chute is actuated using a 360 degree continuous servo, which is controlled using PWM. A digital input of 1 activates and 0 deactivates the servo. The servo uses PWM to reach certain degrees in order to turn the delivery chute towards different delivery containers. This is executed after a bottle has been identified in the testing chamber. The default position, or 0 degree position, of the delivery chute is set to the delivery container for Eska with cap. Thus, if an Eska with cap is identified, the continuous servo is not activated. After every bottle is delivered, the servo is set back to its default position. Since the continuous servo accumulates an offset in position after every rotation, an analog TCRT5000L IR sensor is used to ensure accurate return of the delivery chute to its default position. When the IR sensor sends a signal to the PIC that the delivery chute has returned to 0 degree position, the PIC deactivates PWM for the servo.

Below is a code snippet that shows the PWM used to turn the delivery chute to the delivery container for Yop with cap along with the PWM used to return the chute to its default position with the help of an IR sensor. Similar code is used to turn the delivery chute to the remaining two containers.

```
readADC(3); //reads analog input from RA3 (IR TCRT5000L) on delivery arm

while ((ADRESH*256 + ADRESL > 0x19)){   //keep rotating delivery chute until TCRT5000L IR sensor
        readADC(3);                     //detects that it has returned to the 0 degree position
        RC7 = 1;                //PWM for continuous servo
        __delay_us(2000);
        RC7 = 0;
        __delay_us(18000);
}
...                                     //loading bin actuation and identification code

if (brand == ESKA && type == CAP && curr_state == OPERATION){
```

```
        openbotlid();                    //no rotation needed
        eska_a++;                        //add to ESKA A count
}
else if (brand == YOP && type == CAP && curr_state == OPERATION){
        for( unsigned int i=0;i<5;i++) {        //PWM for continuous servo, rotate chute to 90 degrees
          RC7 = 1;                              //RC7 is the pin for continuous servo
          __delay_us(2300);
          RC7 = 0;
          __delay_us(17700);
        }
        openbotlid();
        yop_a++;                         //add to YOP A count
}
...
```

9.2.9 LCD User Interface

9.2.9.1 Standby State

Figure 29 shows the screen when the program begins or when it is on standby mode. This screen displays real-time date (YY/MM/DD) and time (HH:MM:SS) read from the Real-Time Clock (RTC) chip on the PIC development board, as well as a user prompt to access a menu.

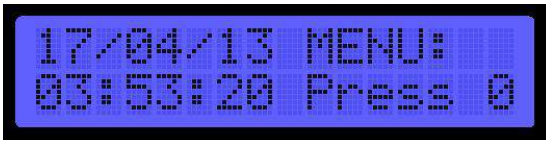
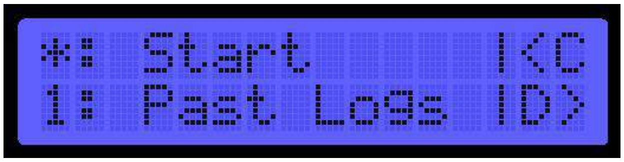


Figure 29. Standby Screen

Once 0 is pressed on the keypad, PIC displays a set of menu options: Start, Past Logs, About and Credits (Figure 30). Keys C and D can be pressed to go back and forth between different screens, respectively.



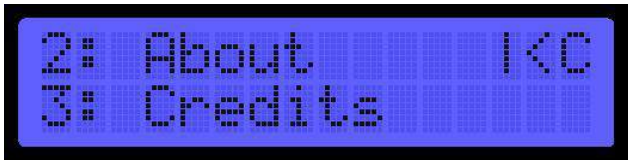
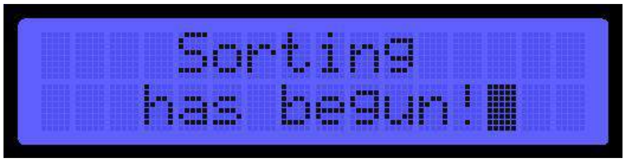


Figure 30. (From Left To Right) A) Menu Part 1 Screen B) Menu Part 2 Screen

Pressing * displays a series of screens as shown in Figure 31 and the program enters the operation phase.



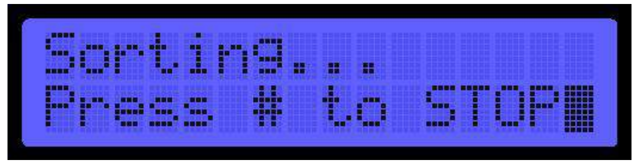
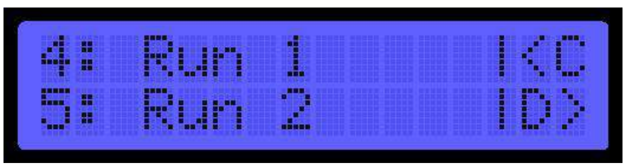
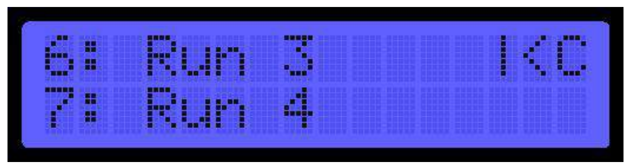


Figure 31. (From Left To Right) A) Entering Operation Screen 1 B) Screen 2

Pressing 1 displays permanent logs for the past four runs stored in EEPROM memory. Pressing 4, 5, 6 or 7 displays a screen similar to Figure 32C, showing the count for Yop with cap (YOP A) or without cap (YOP B), and Eska with cap (ESK A) or without cap (ESK B).





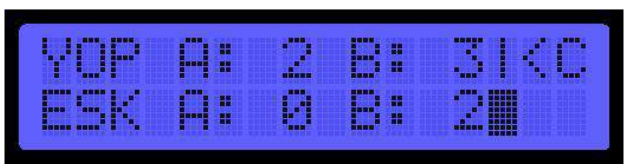


Figure 32. (From Top Left To Right To Bottom) A) Past Logs Part 1 Screen B) Part 2 Screen
C) Sample Past Log Screen (Run 1)

Pressing 2 shows basic information on the machine including the machine name, version and company name (Figure 33).

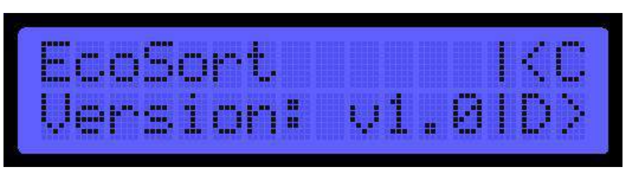


Figure 33. (From Left To Right) A) About Part 1 Screen B) About Part 2 Screen

Pressing 3 gives credits to each subsystem lead: mechanics, circuitry and programming.

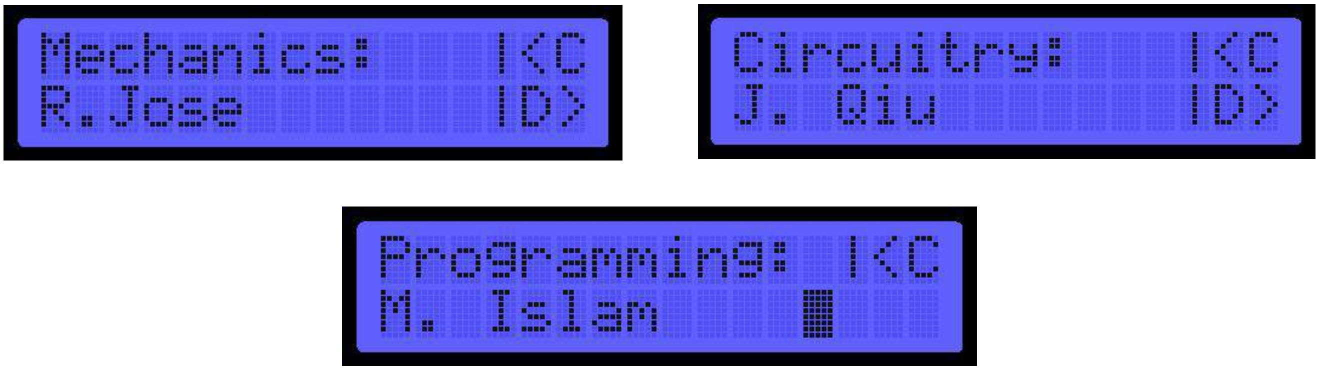


Figure 34. (From Top Left To Right To Bottom) A) Mechanics Credits Screen B) Circuitry Credits Screen C) Programming Credits Screen

9.2.9.2 Operation State

Figure 31B is displayed during the operation. If # is pressed at any point during the operation, the program enters the end operation phase.

9.2.9.3 End Operation State

Once the program enters the end phase, a series of screens (Figure 35) are displayed to notify the user than the operation has terminated and prompts the user for the next step: collect log or return Home or standby mode. If A is pressed, the program enters the collect log state, while if B is pressed, the program goes back to standby state.

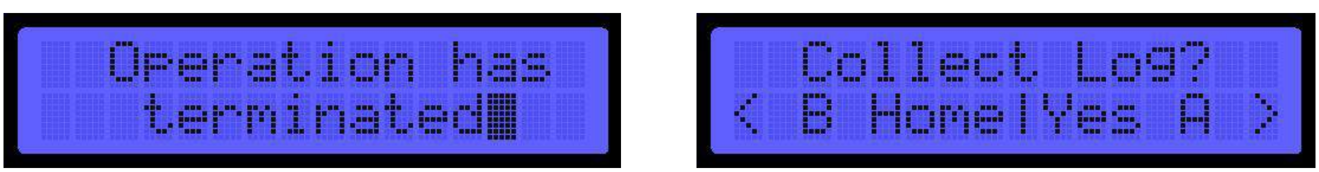


Figure 35. (From Left To Right) A) Termination Message B) Collect Log User Prompt Screen

9.2.9.4 Collect Log State

In the log state, a series of screen (Figure 36) displays the bottles identified in each category, in total and the total operation time for the current run.

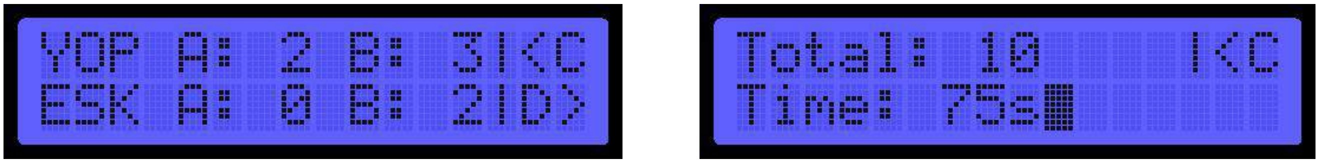


Figure 36. (From Left To Right) A) Current Log Screen B) Total Bottle Count And Operation Time Screen

9.2.10 Keypad Interrupts

The entire program can be interrupted through two keypad inputs: pressing # to terminate operation or B to return to standby mode. Pressing # also stops the motors and returns all actuators to default position before entering log state. Below is the interrupt handler code for keypad interrupt.

```
INT1IE = 1;        //INT1 external interrupt is enabled
ei();        //enable all interrupts
...
void interrupt keypressed(void) {
    if(INT1IF){
        switch(keys[(PORTB & 0xF0) >> 4]){
            case '#':
                update_time(end_time);
                RC7=0; //stop cts servo
                RB0=1; //stop motor
                closebotlid();
                opentoplid();
                lcdclear();
                print(" Operation has  ","   terminated   ");
                delay(1);
                RA0 = 1;
                curr_state = LOG;
                break;
            case 'B':
                curr_state = STANDBY;
                break;
            default:
                break;
```

```
        }
        INT1IF = 0;     //Clear flag bit
    }
}
```

9.3 Suggestions for Improvement of the Subsystem

The microcontroller program code can be optimized for time efficiency. This includes using the PWM feature on the development board to overlap multiple tasks such as closing the bottom lid and opening the top lid at the same time, and rotating the delivery chute and actuating the loading bin at the same time. Currently, the code is sequential and a line of code is executed at a time and before the next line can be executed. Thus, overlapping of multiple actuations can reduce the time taken for operation. Redundant loops can be eliminated through efficient and smart coding, thereby also reducing the time complexity of the code algorithm.

Currently, the loading bin must be adjusted to tilt to its lowest position towards the capture ramp prior to starting the program. Code written to return the loading bin to a default position during the end phase of the program can improve usability of the machine by reducing the calibration and set up needed prior to operation.

PC interface with PIC can be implemented in order to make operation information (time/date, sorting logs) readily downloadable on a PC.

10. INTEGRATION

The first phase of integration was the integration of two out of the three systems -- usually circuitry and electromechanical integration. The next phase involved the integration of third subsystem. The final phase was the debugging of the three integrated subsystems and the mounting of the PIC and other circuit boards onto the machine. The description of the integration process of the three subsystems will be done chronologically for each of the three mechanisms.

10.1 Loading and Orientation Mechanism

Unlike the other two mechanisms, the integration of subsystems for the loading and orientation mechanism started with the integration of the circuitry and the PIC.

The ability to send signals to the motor driver system and control a DC motor was the first step in integrating to form the loading and orientation mechanism. The initial system was the control of a Shenzhen DC gearmotor with the PIC. PWM had not yet been implemented since the motor speed was also dictated by the load on the motor, which required the completion of the mechanical aspect of the loading and orientation mechanism. There was a lot of mechanical redesign with the cradle mechanism, and a placeholder motor was used in order to obtain a better understanding of the range of motion available.

The main redesign was the result of an issue in space and agitation. In the previous design, the cradle was not positioned on top of the delivery containers, but rather off to the side. It was determined that the previous prototype met neither the space constraints nor the ability for agitation required of the mechanism. After prototyping various alternative configurations, the cradle was finally relocated to the top of the delivery containers, which not only solved the space issue but also allowed for a longer cradle that improve orientation and agitation. The entire redesign placed the integration of the mechanism behind the other mechanisms, as the PWM could not be properly tested until the final design, which provided the correct load that was to act on the DC motor.

Once the mechanical redesign had been complete, the integration of circuitry + microcontroller and the electromechanical subsystem was quickly completed. However the debugging process regarding the PWM continued well beyond the initial integration

10.2 Identification Mechanism

The sensors were mounted onto the testing chamber at a relatively early stage of the project. Due to the modular design of the machine, the entire testing chamber was removed to be integrated with the circuitry and sensors. Few problems arose during the integration phase for the identification mechanism. The most prominent and toughest issue was the calibration and configuration of the laser break beam system.

In contrast with the IR break beam system, lasers have a much more focused beam angle. This demanded that the laser and the laser detector be placed very precisely opposite of each other. The initial holes that were drilled into the testing chamber for the laser and detector were off by a few millimetres, and the laser detector failed to activate. To add more complication, the laser was quite loose in its cover and was able to wiggle -- further reducing the reliability. In order to solve this, the wiggle needed to be fixed. Realizing the wiggle of the laser to be too much of a variable, the laser was first epoxied into a fixed position. From there, the laser was turned on and a hole was marked and drilled where the laser shone.

When the PIC was finally integrated with the testing chamber, a few more issues arose.

After fixing the laser and detector into a good position, the laser was capable of penetrating through Eska bottles, thus failing the function as a bottle detection sensor. The solution to this problem involved trial and error; many small-valued resistors ~10 - 200 Ohms added into series the laser beam in a effort to lower the intensity. When the intensity of the laser is lowered, even the slightest scatter by an Eska bottle would be able to trigger the detector. Although this issue still periodically arises, it is now much more uncommon.
Another issue was the interference between the IR sensors. Since there are multiple IR emitters and phototransistor receivers, under the right circumstances, it was possible for one IR emitter to set off an IR receiver from a separate sensor system. In particular, the two TCRT5000 emitters interfered with the Ir receiver of the break beam sensor. The solution to this problem was very simple and drastically reduced the occurrence of this issue; a small piece of wood was attached to the inner roof of the testing chamber -- just before the IR receiver -- which blocks incoming IR.

10.3 Delivery Mechanism

The circuitry was first integrated with the electromechanical subsystem in order to provide information regarding the heights and positions at which the delivery chute was optimal.

The mechanism was then integrated with the PIC microcontroller in order to receive PWM information to rotate the continuous servo which controls the chute by positions on 90 degrees. Since for a continuous servo the PWM controls the speed and not the absolute position of the servo arm, a reference point needed to set and these "discrete" positions were obtained by controlling the PWM as well as the time for which the PWM signal is sent.

The main problem that arose was the unpredictability in the offset and accumulation of offset in the servo position after delivering each bottle. This problem proved difficult to solve and switching to a stepper motor would have been a good solution if time had permitted. A solution to the problem instead involved the addition of another IR reflection sensor. This IR sensor would provide some feedback on the position of the servo, thus the PIC was able to communicate with it based on a reference position. The IR sensor detects when the delivery chute is at the default reference position and sends a signal which stops the servo rotation. With this method, the initial set up procedure is also simplified since the reference position is set automatically. See 8.2.3.2 for more information on this mechanism.

11. BUDGETING

11.1 Electromechanical

Item	Quantity	Cost per unit (CAD)	References
DC Motor	1	15.99	[23]
FS90R Continuous Servo	1	5.18	[24]
FS90 Servo	2	4.35	[25]
Aluminium Sheet 36' x 36'	1	21.98	[26]
1/2'' x 4' x 6' Plywood	1	26.02	[27]
1/4'' x 1' x 1' Plywood	1	2.99	[28]
1/4'' x 48" Metal Rod	1	5.61	[29]
3'' x 1/2' ABS Pipe	1	0.99	[30]
Misc (screws, brakets, washers)	x	10.00	Home Depot
Subtotal		**97.46**	

Note that the prices for common components such as nuts, bolts, rivets and screws cannot be determined at this time. However, total price increase is expected to be negligible due to the low part costs.

11.2 Circuits and Sensors

Item	Quantity	Cost per unit (CAD)	References
TCRT5000 IR Proximity	3	5.1	[31]
Module TCRT5000 IR Proximity	1	1.13	[32]
Laser Detector Module	1	6.50	[33]
Red Laser Pointer	1	3.50	[34]
IR Pair LTE-5208 & LTR-3208	1	1.40	[35]

L298N Dual H-Bridge Motor Driver	1	7.54	[36]
Power Supply 12V DC 3A	1	13.27	[37]
Prototyping Board with Trace 4.2 x 7 cm	1	1.95	[38]
Prototyping Board 5 x 7 cm	1	0.55	[39]
40 Pin IDC Header	1	0.80	[40]
Emergency Stop Button	1	6.25	[41]
Barrel Connectors (M + F)	1	2.80	[42]
Wires and Misc	x	7.00	
Subtotal		**57.57**	

Note that the prices for common components such as wire, solder, LEDs, resistors and capacitors cannot be determined at this time. Additional diodes and inexpensive transistors like the TIP142 are also included in "common components". However, the total price increase is expected to be negligible due to the low part costs.

11.3 Programming

Item	Quantity	Cost (CAD)	References
PIC DevBugger Development Board	1	50.00	Provided by Instructor
LCD+Keyboard	1	6.00	Provided by Instructor
RTC Chip and Coin Battery	1	5.00	Provided by Instructor
PIC18F4620	1	5.37	[43]
PIC18F2550	1	6.50	[44]
Subtotal		**72.87**	

11.4 Budget Summary

Below are the subtotals and total projected project cost in CAD:

Electromechanical	**97.46**
Circuits and Sensors	**57.57**
Programming	**72.87**
Total Projected Cost	**227.90**

The subsystem which is given the greatest share of the budget is the electromechanical subsystem. The majority of the cost is attributed to construction materials. However, the material costs were determined based on the cost of an entire section of the material. Production on a larger scale may help in mitigating the material costs and the cost of the project overall.

The total project cost is 222.90 CAD. This falls below the cost constraint of 230 CAD as stated in the RFP. More exact purchases of construction material and further calculations for optimizing the machine components may be able to decrease this cost even further.

12. WORK SCHEDULE

12.1 Projected Timeline : Gantt Chart

AER201 Project Timeline	start	end
Electromechanical	**01/20/17**	**02/27/17**
Basic Geometry and Schematics	01/20	01/25
Acquire Fabrication Materials	01/26	01/27
Kinematics, Weight, and Moment of Inertia Calculations	01/26	02/03
Detailed Geometry and Schematics	01/26	02/03
Build V-Shaped Cradle	01/31	02/16
Build Sorted Bottle Containment Units	01/31	02/16
Build Testing Chamber and Delivery Arm	01/31	02/16
Attach Testing Chamber and Arm to Base	02/17	02/27
Attach V-Shaped Cradle to Base	02/21	02/27
Circuits and Sensors	**01/20/17**	**02/27/17**
Acquire Power Supplies, Cables, and Sensors	01/20	02/03
Create one 1 H-Bridge Motor Circuit	01/20	02/03
Detailed Analysis of Circuits and Voltages	01/20	01/31
Finalized Power Requirements and Specifications	02/01	02/03
Assemble Motor Circuits on Solder Board	02/06	02/13
Assemble Sensor Circuits on Solder Board	02/06	02/13
Debug all Circuits	02/14	02/20
Solder Circuit Elements onto Protoboard	02/21	02/23
Investigations into the Possible Effects of Signal Interference	02/24	02/27
Microcontroller	**01/20/17**	**02/27/17**
Code for Running Keypad and LCD	01/20	02/02
Pseudo-code for Final Machine Program	01/20	02/02
Code for Handling Interrupts (General, Specific)	01/20	02/08
Code for V-Shaped Bin Actuation (w/ Motors)	02/09	02/23
Code for Delivery Arm Actuation (w/ Motor)	02/09	02/23
Code for Sensors (IR Proximity, Light/Colour)	02/09	02/23
Code for Testing Chamber Lids Actuation (w/ Servo Motors)	02/09	02/23
Code for LCD User Interface (Menu, Termination Message)	02/09	02/23
Code for Emergency Stop	02/09	02/23
Code for Real Date/Time Display	02/17	02/23
Code for Sorting Logs	02/17	02/23
Debug all Code and Download onto Board	02/08	02/27
Integration	**02/28/17**	**04/10/17**
Mount Microcontoller Board	03/09	03/16
House Power Supply and Sensors on Machine	02/28	03/13
Integrate Circuits and Code	02/28	03/27
Test Mechanical Motions	02/28	03/30
Place Sensors in Appropriate Locations on Machine	02/28	03/21
Test Complete Machine and Debug	03/17	04/10
Evaluations	**02/07/17**	**04/11/17**
Interim Notebook Evaluation	02/15	02/15
Individual Performance Evaluations: Fabrication and Integration	02/07	02/07
Individual Performance Evaluations: Functionality and Integration	02/28	02/28
Team Evaluation: Functionality and Integration	03/14	03/14
Team Evaluation: Functionality and Debugging	03/28	03/28
Public Demonstration	04/11	04/11

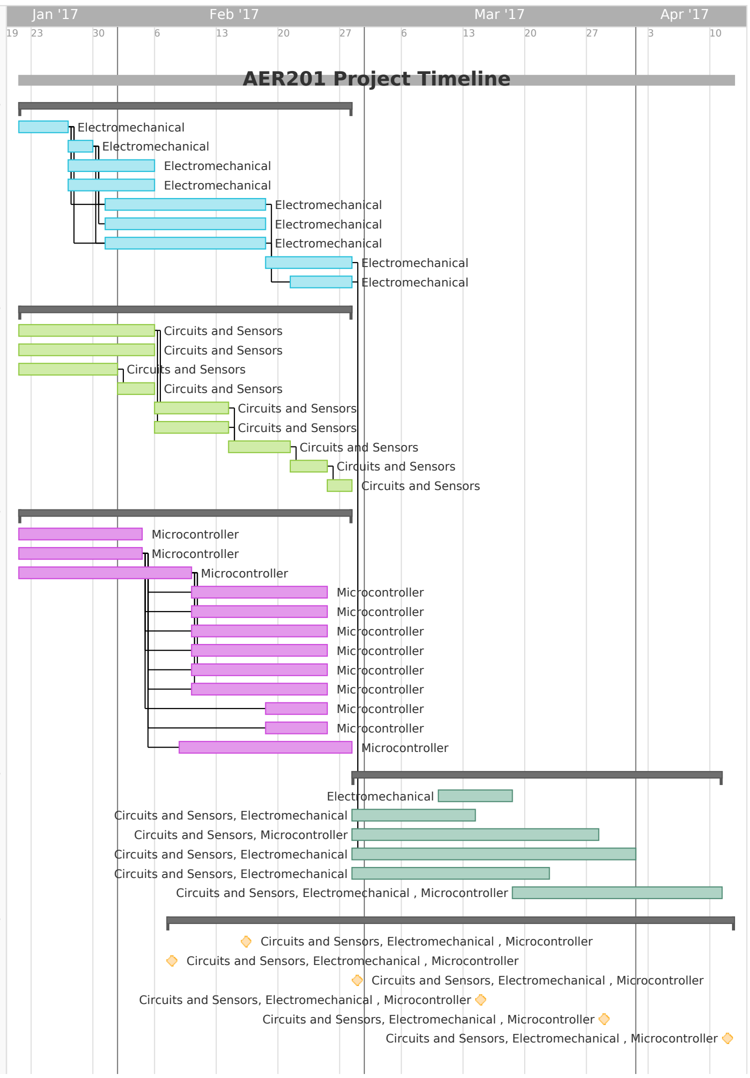


Figure 31 Gantt Chart Showing Task Distribution And Timeline For The Team For The Duration Of The Project

12.2 Actual Timeline : Modified Gantt Chart

AER201 Project Timeline	start	end
Electromechanical	**01/23/17**	**02/27/17**
Basic Geometry and Schematics	01/23	01/25
Acquire Fabrication Materials	01/26	01/27
Kinematics, Weight, and Moment of Inertia Calculations	01/26	02/03
Detailed Geometry and Schematics	01/26	02/06
Build V-Shaped Cradle	01/31	02/14
Build Sorted Bottle Containment Units	01/31	02/10
Build Testing Chamber and Delivery Arm	01/31	02/16
Attach Testing Chamber and Arm to Base	02/17	02/21
Attach V-Shaped Cradle to Base	02/21	02/27
Circuits and Sensors	**01/20/17**	**03/22/17**
Detailed Analysis of Circuits and Voltages	01/20	01/31
Acquire Power Supplies, Cables, and Sensors	01/20	02/03
Create one 1 H-Bridge Motor Circuit	01/20	02/03
Finalized Power Requirements and Specifications	02/01	02/03
Assemble Motor Circuits on Solder Board	02/06	02/13
Assemble Sensor Circuits on Solder Board	02/06	02/13
Debug all Circuits	02/14	02/20
Solder Circuit Elements onto Protoboard	02/21	03/21
Investigations into the Possible Effects of Signal Interference	02/24	03/22
Microcontroller	**01/19/17**	**04/10/17**
Code for Real Date/Time Display	01/19	01/27
Code for Running Keypad and LCD	01/20	02/02
Pseudo-code for Final Machine Program	01/20	02/02
Code for Handling Interrupts (General, Specific)	01/20	02/08
Code for Emergency Stop	02/09	02/23
Code for LCD User Interface (Menu, Termination Message)	02/09	03/08
Code for Testing Chamber Lids Actuation (w/ Servo Motors)	02/09	03/07
Code for V-Shaped Bin Actuation (w/ Motors)	02/09	03/10
Code for Delivery Arm Actuation (w/ Motor)	02/09	03/10
Code for Sensors (IR Proximity, Light/Colour)	02/09	03/17
Debug all Code	02/08	04/10
Code for Sorting Logs	02/17	03/31
Integration	**02/28/17**	**04/10/17**
Integrate Circuits and Code	02/28	04/10
Test Mechanical Motions	02/28	04/10
Create Modified V-Shaped Cradle	02/28	03/13
Solder all Remaining Circuits	02/28	03/14
Place Sensors in Appropriate Locations on Machine	02/28	03/21
Create Capture Ramp After Testing	03/06	03/28
Test Complete Machine and Debug	03/17	04/07
Mount Microcontroller and Circuit Boards	03/24	03/27
House Power Supply and Sensors on Machine	04/04	04/07
Evaluations	**02/07/17**	**04/11/17**
Interim Notebook Evaluation	02/15	02/15
Individual Performance Evaluations: Fabrication and Integration	02/07	02/07
Individual Performance Evaluations: Functionality and Integration	02/28	02/28
Team Evaluation: Functionality and Integration	03/14	03/14
Team Evaluation: Functionality and Debugging	03/28	03/28
Public Demonstration	04/11	04/11

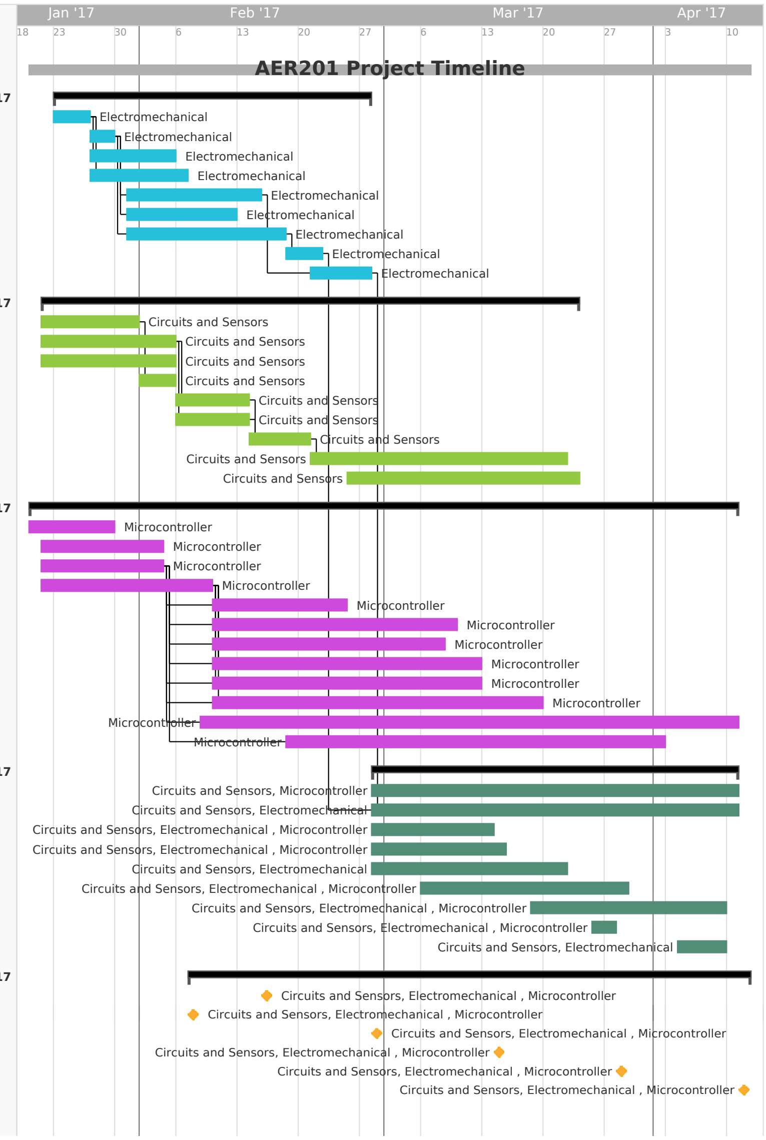


Figure 31 Modified Gantt Chart Showing Actual Task Distribution And Timeline For The Team For The Duration Of The Project

As anticipated, at the integration phase, our team encountered problems that lead to the re-design of certain components of the machine. The major lapse was that the machine was just slightly outside of the dimensional constraints. In addition, the V-shaped cradle that we built was too small and thus caused major clogging when it was filled with the maximum 10 bottles.

The team then had to re-focus our collective attention on the electromechanical parts that had to be remade or changed, to ensure that we would have ample time for testing and experimenting. Clearly, once the integration began, our team chose to work very close with each other. This helped us to learn from and provide support to each other, while creating a highly functional and accurate sorting machine.

13. SYSTEM IMPROVEMENT SUGGESTIONS

The current design of the machine allows for improvement in the following three areas:

1. Safety
2. Reliability and jam prevention
3. Compatibility of power supply and components

Based on experimentation, the major safety concern is the ejection of bottles from the loading bin. This occurrence is closely linked to jamming at the opening of the cradle. The walls of the capture ramp have been extended in order to tackle this issue, but despite this fix, there have been observed cases of bottles being ejected.

Secondary agitation mechanisms would improve the safety as well as the reliability of the machine through jam prevention. Improvement options of this type, that have been considered include vibration pads on the cradle to facilitate release, and rotating arms from a motor to push back bottles aggregated at the jamming site.

The other potential for improvement lies in the power supply for the machine. A 5 V power supply in addition to a 12 V power supply would help this machine to become more efficient in terms of power consumption. The machine currently uses only a 12 V supply, and power is being lost as heat at the voltage regulators.

14. CONCLUSION

Overall, the project was completed successfully. The machine accommodates the given constraints and operates reliably. The solution is an inexpensive and reliable autonomous robot which automates part of the recycling process, namely the classification of certain recyclable products. This machine is designed to sort two types of bottles, Yop and Eska, and can also sort based on the presence of a bottle cap. The design is simple, using minimal actuators and avoiding complicated mechanical movements.

Due to these features, the machine qualified for the AER201 competition, with one successful trial run wherein it accurately sorted 8 out of 9 bottles.

The cost and time constraints imposed on the development of this machine restrict the diversity of bottles that can be sorted. With additional resources, future developments of this machine through the inclusion of more advanced mechanical parts and sensors can allow for a much more diverse set of recyclables to be sorted with reduced time and labour costs. In addition, the modularity of the design allows for easy customization — each of the three mechanisms can be developed separately and incorporated into other machines. This machines automation of the recycling process can make recycling a more widespread and prominent phenomenon.

REFERENCES

[1] M. R. Emami, Multidisciplinary Engineering Design from Theory to Practice, 2017th ed. Canada: McGraw-Hill Education, 2017.

[2] "Bottle unscrambling machine/bottle sorting machine," in YouTube, YouTube, 2015. [Online]. Available: https://www.youtube.com/watch?v=VoAjTI4twdg&index=3&list=PLgWoRCEfssa7HQrJ0uMW_tWSdYTpP4j7h. Accessed: Feb. 1, 2017.

[3] Dyco, "Vision guided robotic bottle orienting system for Blowmold take-away — Dyco Inc," 2017. [Online]. Available: http://www.dyco-inc.com/Robotic-Blowmold-Take-Away-(VGRBO). Accessed: Feb. 1, 2017.

[4] "Bottle unscrambler, bottle sorting machines," BCM Engineering. [Online]. Available: http://www.bcm-engineering.com/EN/products/plastic-bottles-unscrambler-machines.html. Accessed: Feb. 1, 2017.

[5] "Plastic bottles orienter machines," BCM Engineering. [Online]. Available: http://www.bcm-engineering.com/EN/products/plastic-bottles-orienter-machines.html. Accessed: Feb. 1, 2017.

[6] "Automatic plastic bottle Unscrambler/sorting machine," 2014. [Online]. Available: http://www.links-machine.net/sell-1135773-automatic-plastic-bottle-unscrambler-sorting-machine.html. Accessed: Feb. 1, 2017.

[7] "Automatic plastic bottle sorting machine pet bottle Unscrambler with elevator system," in YouTube, YouTube, 2014. [Online]. Available: https://www.youtube.com/watch?v=pN0sZZuAHVQ. Accessed: Feb. 1, 2017.

[8] Corporation2017Pololu, "Sharp GP2Y0A51SK0F analog distance sensor 2-15cm," 2001. [Online]. Available: https://www.pololu.com/product/2450. Accessed: Feb. 1, 2017.

[9] C. Inc, D. L. Electronics, and Robotics, "940nm infrared Emitter & transistor pair (LTE-302 & LTR-301)," Creatron. [Online]. Available: https://www.creatroninc.com/product/940nm-infrared-emitter-transistor-pair-lte-302-ltr-301/. Accessed: Feb. 1, 2017.

[10] C. Inc, D. L. Electronics, and Robotics, "Laser detector," Creatron. [Online]. Available: https://www.creatroninc.com/product/laser-detector/?search_query=laser+emitter&results=12. Accessed: Feb. 1, 2017.

[11] C. Inc, D. L. Electronics, and Robotics, "Red laser pointer - 3.3 to 5V," Creatron. [Online]. Available: https://www.creatroninc.com/product/red-laser-pointer-33-to-5v/?search_query=laser+emitter&results=12. Accessed: Feb. 1, 2017.

[12] . [Online]. Available: https://www.creatroninc.com/product/analog-digital-light-sensor/. Accessed: Feb. 1, 2017.

[13] A. Industries, "RGB color sensor with IR filter and white LED - TCS34725 ID: 1334 - $7.95: Adafruit industries, unique & fun DIY electronics and kits,". [Online]. Available: https://www.adafruit.com/product/1334. Accessed: Feb. 1, 2017.

[14] "Aladdin™ is the most popular optical Sorter for plastic by MSS, Inc," 2012. [Online]. Available: http://www.mssoptical.com/material-sorting-equipment/plastic-sorting-equipment/aladdin/. Accessed: Feb. 1, 2017.

[15] "Skittles M&M's sorting machine - ivc wiki,". [Online]. Available: http://beta.ivc.no/wiki/index.php/Skittles_M%26M%27s_Sorting_Machine. Accessed: Feb. 1, 2017.

[16] "53 automatic bottle-sorting system - final video," in YouTube, YouTube, 2012. [Online]. Available: https://www.youtube.com/watch?v=OzPAkK031UU. Accessed: Feb. 1, 2017.

[17] "Arduino - ArduinoBoardMega2560", Arduino.cc, 2017. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardMega2560. [Accessed: 13- Apr- 2017].

[18] I. Yani and I. Budiman, "Development of identification system of cans and bottle," Journal of Physics: Conference Series, vol. 622, p. 012053, Jun. 2015.

[19] D. A. Wahab, A. Hussain, E. Scavino, M. M. Mustafa, and H. Basri, "Development of a prototype automated sorting system for plastic recycling," American Journal of Applied Sciences, vol. 3, no. 7, pp. 1924–1928, 2006. [Online]. Available: https://pdfs.semanticscholar.org/226b/84079f957b13a1391fa27d4e496108608151.pdf. Accessed: Feb. 1, 2017.

[20] H. Frankel, S. Miroshnichenko, J. B. Whitlock, Rutgers, and T. State, Patent US5314072 - sorting plastic bottles for recycling. Google Books, 1992. [Online]. Available: https://www.google.com/patents/US5314072. Accessed: Feb. 1, 2017.

[21] F. G. Thompson, L. P. Vogel, E. R. Wagner, T. F. G, V. L. P, and W. E. R, Patent US4248389 - container sorting and handling system. Google Books, 1979. [Online]. Available: https://www.google.com/patents/US4248389. Accessed: Feb. 1, 2017.

[22] N. Dumont, "nathandumont.com : H-Bridge Tutorial", Nathandumont.com, 2017. [Online]. Available: http://nathandumont.com/blog/h-bridge-tutorial. [Accessed: 13- Apr- 2017].

[23]" 120:1 25D Metal Gear Motor (6V 50RPM)", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/120-1-25d-metal-gear-motor-6v-50rpm/. [Accessed: 13- Apr- 2017].

[24] "Pololu - FEETECH FS90R Micro Continuous Rotation Servo", Pololu.com, 2017. [Online]. Available: https://www.pololu.com/product/2820. [Accessed: 13- Apr- 2017].

[25] "Pololu - FEETECH FS90 Micro Servo", Pololu.com, 2017. [Online]. Available: https://www.pololu.com/product/2818. [Accessed: 13- Apr- 2017].

[26] "M-D Building Products 36 in. x 36 in. Plain Aluminum Sheet in Silver-57000 - The Home Depot", The Home Depot, 2017. [Online]. Available: http://www.homedepot.com/p/M-D-Building-

Products-36-in-x-36-in-Plain-Aluminum-Sheet-in-Silver-57000/100351161. [Accessed: 13- Apr- 2017].

[27] "Sande Plywood (Common: 1/2 in. x 4 ft. x 8 ft.; Actual: 0.472 in. x 48 in. x 96 in.)-454532 - The Home Depot", The Home Depot, 2017. [Online]. Available: http://www.homedepot.com/p/Sande-Plywood-Common-1-2-in-x-4-ft-x-8-ft-Actual-0-472-in-x-48-in-x-96-in-454532/100017950. [Accessed: 13- Apr- 2017].

[28] "Sande Plywood (Common: 1/4 in. x 4 ft. x 8 ft.; Actual: 0.205 in. x 48 in. x 96 in.)-479023 - The Home Depot", The Home Depot, 2017. [Online]. Available: http://www.homedepot.com/p/Sande-Plywood-Common-1-4-in-x-4-ft-x-8-ft-Actual-0-205-in-x-48-in-x-96-in-479023/100073744. [Accessed: 13- Apr- 2017].

[29] "Everbilt 1/4 in. x 48 in. Aluminum Round Rod-800377 - The Home Depot", The Home Depot, 2017. [Online]. Available: http://www.homedepot.com/p/Everbilt-1-4-in-x-48-in-Aluminum-Round-Rod-800377/204604769. [Accessed: 13- Apr- 2017].

[30] "IPEX HomeRite Products ABS PIPE 3 inches x 12 ft CELL CORE | The Home Depot Canada", Homedepot.ca, 2017. [Online]. Available: https://www.homedepot.ca/en/home/p.abs-pipe-3-inches-x--12-ft-cell-core.1000120780.html. [Accessed: 13- Apr- 2017].

[31] "TCRT5000L IR Reflectance Sensor (2.5mm)", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/tcrt5000l-ir-reflectance-sensor-25mm-a/. [Accessed: 13- Apr- 2017].

[32] "IR Reflective Sensor - TCRT5000 [TCRT5000 Modules] - US $0.85 : HAOYU Electronics : Make Engineers Job Easier", Hotmcu.com, 2017. [Online]. Available: http://www.hotmcu.com/ir-reflective-sensor-tcrt5000-p-184.html. [Accessed: 13- Apr- 2017].

[33] "Laser Detector Module", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/laser-detector-module/. [Accessed: 13- Apr- 2017].

[34] "Red Laser Pointer - 3.3 to 5V", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/red-laser-pointer-33-to-5v/. [Accessed: 13- Apr- 2017].

[35] "940nm Infrared Emitter & Transistor Pair (LTE-5208 & LTR-3208)", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/940nm-infrared-emitter-transistor-pair-lte-5208-ltr-3208/. [Accessed: 13- Apr- 2017].

[36] "L298N Dual H Bridge DC Stepper Motor Driver Module Controller Board For Arduino by Mmm999", Tindie, 2017. [Online]. Available: https://www.tindie.com/products/mmm999/l298n-dual-h-bridge-dc-stepper-motor-driver-module-controller-board-for-arduino/. [Accessed: 13- Apr- 2017].

[37] "12VDC 3A Wall Adapter Power Supply", Robotshop.com, 2017. [Online]. Available: http://www.robotshop.com/ca/en/12vdc-3a-wall-adapter-power-supply.html. [Accessed: 13- Apr- 2017].

[38] "4.2 x 7cm Prototyping Board with Trace", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/42-x-7cm-prototyping-board-with-trace/. [Accessed: 13- Apr- 2017].

[39] "5 x 7cm Prototyping Board", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/5-x-7cm-prototyping-board/. [Accessed: 13- Apr- 2017].

[40] "40 Pin (20x2) IDC Shrouded Header", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/40-pin-20x2-idc-shrouded-header/. [Accessed: 13- Apr- 2017].

[41] "Emergency Switch", Creatron Inc, 2017. [Online]. Available: https://www.creatroninc.com/product/emergency-switch/. [Accessed: 13- Apr- 2017].

[42] "2.1mm Barrel Jack to terminal", Robotshop.com, 2017. [Online]. Available: http://www.robotshop.com/ca/en/barrel-jack-terminal-fit0151.html. [Accessed: 13- Apr- 2017].

[43]"PIC18F4620-I/P Microchip Technology | Integrated Circuits (ICs) | DigiKey", Digikey.ca, 2017. [Online]. Available: https://www.digikey.ca/product-detail/en/microchip-technology/PIC18F4620-I-P/PIC18F4620-I-P-ND/613256. [Accessed: 13- Apr- 2017].

[44] "PIC18F2550-I/SP Microchip Technology | Integrated Circuits (ICs) | DigiKey", Digikey.ca, 2017. [Online]. Available: https://www.digikey.ca/product-detail/en/microchip-technology/PIC18F2550-I-SP/PIC18F2550-I-SP-ND/704575. [Accessed: 13- Apr- 2017].

APPENDICES

APPENDIX A: STANDARD OPERATING PROCEDURE

SetUp

1. Set the machine down on a level surface and plug the power supply into a wall outlet.
2. Ensure that the opening of the loading bin is tilted to its lowest point, towards the capture ramp.
3. Load the bin with a maximum of 10 plastic bottles. The supported type of bottles are 200-ml white HDPE plastic bottles (e.g. Yop brand) with (A) or without cap (B), and 330-ml transparent PET plastic bottles (e.g. Eska brand) with (A) or without cap (B).
4. Turn on the ON/OFF switch on the PIC DevBugger development board.

Operation

1. Press 0 to display the menu.
2. Press * to start the operation. During operation, exercise caution by remaining outside the parameters of the machine.
3. Press # to terminate the operation at any point. Otherwise, wait till the operation is completed.
4. Collect the sorted bottles from the removable delivery containers.
5. Turn off the ON/OFF switch on the PIC DevBugger development board.

Keys

Standby Prompt Screen

0: Menu
*: Start Operation
1: Past Logs
    4: Run 1
    5: Run 2
    6: Run 3
    7: Run 4
2: About
3: Credits
C: Back
D: Next

In-Operation Prompt Screen

#: Stop Operation

Collect Current Log Prompt Screen

A: Collect Current Log
B: Return to Standby Mode
C: Back
D: Next

CAUTION:

In case of emergency, press the red emergency button located on the side of the machine.

APPENDIX B: DECISION MAKING

B-1 Alternative Solutions

Loading and Orientation Designs

During the initial phase, three major types of designs were conceptually developed for loading and orientation. This section outlines the different variations within each type of design that were explored.

Funnels

Figure 14 shows a vertical funnel the wheels are stationary and the door is closed during the loading period. Once the machine is started, the wheels are activated to prevent bottles from jamming at the neck of the funnel and allow them to fall vertically into the narrow stem. A sensor is attached midway against the inner wall of the stem to detect the presence of a bottle so that the door can be signalled to open in order to allow the bottles through one at a time. Figure 15 shows a tilted half-funnel consisting of automated swinging door, a vibration motor and a IR proximity sensor. For this design, the funnel is stationary and the door is closed during the loading period. Once the machine is started, the motor causes the funnel to vibrate and the vibration motion slides the bottles towards the narrow stem. Once the proximity sensor detects a bottle in the stem, the door opens to allow the bottle through horizontally.

The major problem with funnel designs is that the transition between the funnel body and stem creates a bottleneck where two bottles tend to want to slide through at the same time, thus jamming the entrance to the funnel stem.

Centrifuges

The idea of centrifugal loading bins was initially inspired by a past AER201 project on vial sorting [21]. Figure 16 illustrates a centrifugal loading bin, which can either be straight upright or tilted slightly, with an automated door (or a mechanical flap door) which is attached to a ramp. Once the machine is started, the loading bin spins and the door opens, allowing bottles to pass through horizontally and roll along the ramp to the next stage of the sorting process. Figure 18 shows another centrifugal loading bin that consists of several depressed wells containing trap doors along its wall. There is a vertical testing chamber connected to each trap door. Once the machine starts, the loading bin spins and the centrifugal force pushes bottles towards the wall of the bin where they fall into the depressed wells. A proximity sensor is placed in each well and once a bottle is detected in a well, the trapdoor opens allowing the bottle to fall through vertically into the testing chamber. Figure 17 shows a similar design, but this time with a tilted loading bin consisting of several holes stacked onto of a lower base consisting of one hole which has an attached testing chamber. Once the machine is started, the loading bin spins and bottles drop into the holes along the walls. Since the bin keeps rotating, one of the bottles in one of the holes will fall horizontally into the testing chamber when a hole on the loading bin aligns with the hole on the bottom base. The cone in the middle is stationary and prevents bottles from getting stuck at the centre of the bin by sliding them towards the bin wall. This design is inspired by the industrial machine discussed in Section 3.1.4.1 (Figure 4).

The major problem with funnel designs is that a bottle can jam the centrifuge if it does not fall through exit doors or enter into the test chamber properly. If the centrifuge gets jammed, none of the other bottles will be able to move past the initial stage of orientation.

V-Shaped Ramps and the Evolution of the V-Shaped Cradle

The proposed orientation design, the V-shaped cradle, was developed through a series of other V-shaped ramps. Initially, the ramp consisted of only a front wall containing an automated door (Figure 19), which was later changed to remove the front wall, including the automated door, completely. This new design was attached to a half-funnel similar to the one in Figure 15 to guide bottles one by one into a testing chamber (Figure 20). In this design, the automated door was relocated to the end of the testing chamber. Finally, the complete V-shaped cradle emerged (Figure 21) with new modifications to the previous V-shaped ramp design. This design is discussed more in detail in Section 5.1.

Identification Designs

Mechanical Means of Identification

Different mechanical means of identifying the type of bottles were developed. One such method is through the use of a pair of mechanical arms holding rods. If one of the rods is able to enter the bottle, the bottle is identified as uncapped. Otherwise, it is identified as capped. Another mechanical system can consist of multiple doors with different dimensions. Only a certain brand of bottle can pass through a certain door. Once the bottle passes through the door, a proximity sensor can acknowledge that a bottle has passed through a specific door, thus identifying the brand of the bottle.

Testing Chamber with Sensors (Proximity/Colour)

Several types of testing chambers, horizontal and vertical, were developed with different sensor types and their placements. Figure 23 shows an example of a vertical testing chamber that was developed during the initial phase. This design used 8 pairs of sensors, both proximity and colour sensors to identify the type of bottle. The flowchart in Figure 26 outlines the identification algorithm for this preliminary design. Due to the degree of freedom in the chamber to allow both Eska and YOP bottles to fit, combined with the upright position of the chamber, the bottle tended to sit in the chamber at a diagonal instead of completely vertical. This would create misalignment with the sensor positions, thus preventing accurate measurements. As well, this identification algorithm consisted of a lot of steps and it was simplified to the proposed design which consists of a tilted testing chamber with fewer sensors (only proximity). The new design is discussed more in detail in Section 5.2.

Sorting and Delivery Designs

Rotating Bins

One design concept of the delivery mechanism involves a rotating group of the four classification bins (Figure 25). This system was designed to be very simple; a single servo motor will be capable of actuating the four bins, orientating the correct bin below the identification mechanism. The identification chamber can then simply dispense the bottle into the bin below by opening the bottom lid.

Automated Chute

Another design involves an intermediate chute between the identification mechanism and the bins below (Figure 24). In this design, the bins and the identification mechanism both remain stationary while a chute in the middle of the four bins will rotate via a servo in order to direct the bottle into the correct bin based on input from identification.

B-2 Decision Making

The orientation mechanism will be evaluated first, using utility-based and Analytical Hierarchy Process (AHP) decision-making aids. Through prototyping, the orientation step was determined to be the most difficult and complex system to design. The purpose of the orientation mechanism is to output the unsorted group of bottles in an orderly and sequential manner -- either having the bottles side-to-side (sideways) or top-to-bottom (lengthwise).

Jamming prevention and separation mechanisms are considered subsections of the orientation mechanism. They are evaluated after the main orientation mechanism.

Design of the identification mechanism depends on the orientation of the bottles exiting the orientation mechanism. The main design decisions of the identification mechanism involve the placement and the types of sensors that are used. This will be discussed next.

The delivery mechanism has only two main designs, and is evaluated after the orientation and identification mechanisms.

Utility Function and AHP Analysis for the Orientation Mechanism

The three main concepts for the orientation mechanism are the funnel with wheels, the centrifuge, and the V-shaped actuated cradle. Six objectives were selected to help analyze each of the design ideas through a utility-based decision making process.

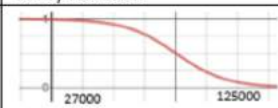
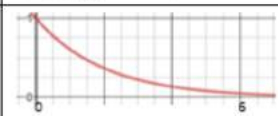
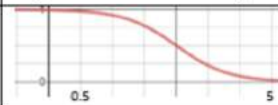
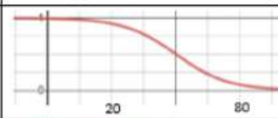
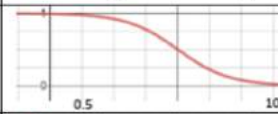
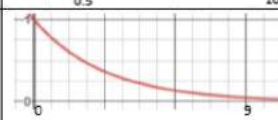
Objective	Parameter	Scale	Unit	Utility Function	Equation of Function
Takes up little space	Smallest volume in which the mechanism can be enclosed approximated by a rectangular prism.	Ratio	cubic centimeters		$U(x) = \frac{1}{1 + e^{\frac{1}{10000}(x-76000)}}$
Requires few actuators	Number of actuators required for the mechanism	Ratio	actuators		$U(x) = e^{-\frac{1}{2}x}$
Lightweight	Weight of mechanism	Ratio	kilograms		$U(x) = \frac{1}{1 + e^{2(x-2.25)}}$
Inexpensive	Cost of material for one copy of mechanism	Ratio	CAD $		$U(x) = \frac{1}{1 + e^{\frac{1}{10}(x-50)}}$
Outputs bottles quickly	Average time required to output one bottle with continuous operation	Ratio	seconds		$U(x) = \frac{1}{1 + e^{(x-5.25)}}$
Infrequent jamming	Average number of jams per 10 trials. For the cradle or centrifuge, a "jam" occurs when no bottles exit the mechanism after 10 oscillations	Ratio	jams		$U(x) = e^{-0.3x}$

Table 1. Objectives, Metrics And Utility Values For Orientation Mechanisms

Each of the six objectives along with their metrics and utility functions, U(x), are displayed in Table 1.

Estimated values for each of the designs for each objective are used as input to the utility function. The utility function outputs a utility value; a dimensionless quantity that ranges from 0 to 1 -- values closer to 1 being more preferred. Weights are also assigned to each objective, as some objectives are valued to be important than others. Utility values, weighted sums, and normalized totals for each design are displayed in Table 2.
For brevity, each design concept will be denoted by a letter:

- A = Funnel with wheels
- B = Centrifuge
- C = V-shaped Actuated Cradle

From utility function analysis, it appears that Design C > Design B > Design A in order of preference. The main advantage of the actuated cradle, Design C, is that the frequency of jamming is relatively low. For most other objectives, Design C is average-above average based on estimated values.

However, there are large uncertainties attributed with utility-based decision making; uncertainty in estimated values, uncertainty in weight assignments, and uncertainty in the curvature of the utility function are the main

D.R. Anderson,
Western Educat

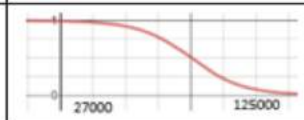
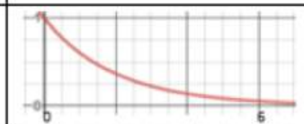
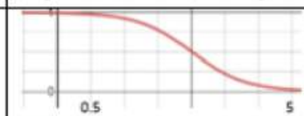
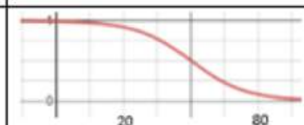
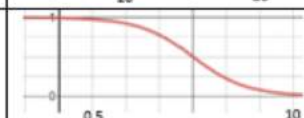
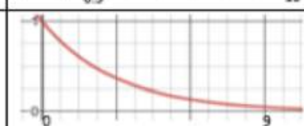
		Design A		Design B		Design C		
Objectives	Weight	Estimated Value	Utility Value	Estimated Value	Utility Value	Estimated Value	Utility Value	Utility
Takes up little space	3	70000	0.65	40000	0.97	50000	0.93	
Requires few actuators	1	3	0.22	1	0.61	1	0.61	
Lightweight	2	1	0.92	2	0.62	1.5	0.82	
Inexpensive	1	50	0.5	60	0.27	50	0.5	
Outputs bottles quickly	2	1.5	0.98	2	0.96	4	0.78	
Infrequent jamming	5	4	0.30	3.1	0.39	1.5	0.64	
Utility Weighted Sum		1.33		1.48		1.72		
Normalized Total		0.29		0.33		0.38		

Table 2. Utility Values And Normalized Totals For Each Design

examples. Since the orientation mechanism is deemed the most critical of the three main mechanisms, decision making through the Analytical Hierarchy Process (AHP) is also performed.

AHP will deal with relative preferences and relative importances which will be assessed quantitatively. A value between 1 - 9 is assigned with the convention in Figure 27.

First, for each objective, the relative preferences of the three design concepts are assigned. These values are displayed in the form of matrices. For example: a value of 3 row A, column B means that Design A is moderately preferred over Design B. Reciprocals are also allowed: a value of ⅕ in row A, column C means that Design C is strongly preferred over Design A.

Matrices for the relatives preferences of the three designs with respect to the six objectives are displayed below along with their consistency ratios (CR). CR < 0.10 generally implies a good consistency.

For brevity, each objective will be assigned a number:

- Obj. 1 = Takes up little space
- Obj. 2 = Requires few actuators
- Obj. 3 = Lightweight
- Obj. 4 = Inexpensive
- Obj. 5 = Outputs bottles quickly
- Obj. 6 = Infrequent jamming

	Obj. 1			CR
	A	B	C	0.003846
A	1	1/5	1/3	
B	5	1	2	
C	3	1/2	1	

	Obj. 2			CR
	A	B	C	0
A	1	1/3	1/3	
B	3	1	1	
C	3	1	1	

	Obj. 3			CR
	A	B	C	0.027885
A	1	6	5	
B	1/6	1	1/2	
C	1/5	2	1	

	Obj. 4			CR
	A	B	C	0.002885
A	1	5	2	
B	1/5	1	1/3	
C	1/2	3	1	

	Obj. 5			CR
	A	B	C	0.0375
A	1	3	5	
B	1/3	1	3	
C	1/5	1/3	1	

	Obj. 6			CR
	A	B	C	0.057692
A	1	1/5	1/7	
B	5	1	1/3	
C	7	3	1	

Table 3. Relative Preferences Of Designs With Respect To Objectives

Normalizing the columns allows for the values in each row to be summed up. The sum of each row provides the overall preference of the design solution associated with that row with respect to the corresponding objective. The normalized matrices and the overall preferences are displayed below.

In addition to the relative preferences of design concepts with respect to objectives, the relative importances of each objective must also be factored into the AHP process:

Normalizing, and calculating of the overall importance of each objective:

Relative Importance of Objectives							CR
	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Obj. 5	Obj. 6	0.091
Obj. 1	1	8	3	5	3	1/5	
Obj. 2	1/8	1	1/3	1/2	1/4	1/9	
Obj. 3	1/3	3	1	2	1	1/7	
Obj. 4	1/5	2	1/2	1	1/2	1/8	
Obj. 5	1/3	4	1	2	1	1/7	
Obj. 6	5	9	7	8	7	1	

Normalized Obj. 3				Overall Preference
	A	B	C	
A	0.73	0.67	0.77	2.17
B	0.12	0.11	0.08	0.31
C	0.15	0.22	0.15	0.52

Normalized Obj. 6				Overall Preference
	A	B	C	
A	0.08	0.05	0.10	0.22
B	0.38	0.24	0.23	0.85
C	0.54	0.71	0.68	1.93

Table 4. Normalization Of Matrices In Figure 14, Along With Overall Preference Of Each Design

Finally, in order to compare the three design concepts, the decision value D for each design must be calculated. The decision value of a given design is the sum of the product of the overall preferences of the design and the corresponding overall importance of the objective. This is given by the formula :

$$D_N = \sum_{M=1}^{6} P_{N_M} * I_M$$

Using the preferences/importance values obtained from the normalized matrices:

- Design A = 3.87
- Design B = 5.87
- Design C = 8.26

Similar to the utility-based analysis, AHP suggests that Design C > Design B > Design A as well. Since both decision-making methods suggest the same results, Design C, the V-shaped actuated cradle, is selected to be the orientation mechanism of the design concept.

Decision Making for Jamming Prevention and Separation of Bottles

The V-shaped actuated cradle was first prototyped using cardboard. With 10 assorted Eska and Yop bottles loaded into the mechanism, 8/10 trials on average were able to dispense all 10 bottles without any jams. When jams occurred, however, the bottles almost always took on a configuration in which they would lie sideways to the exit; neither the top nor the bottom of the bottle was pointed toward the exit. Since this obstructing orientation of bottles occurs at the exit mouth of the cradle, agitation mechanisms that are localized to the exit mouth should be sufficient to decrease the likelihood of jams. Large-scale agitation mechanisms such as vibrating agitators would have to be capable of moving the entire cradle. This may raise power and safety concerns that are described in the RFP.

Two mechanisms for localized agitation were conceived: motorized wheels and linear solenoids actuators. The motorized wheels would be similar to its implementation in Figure 14. In addition to directing bottles through the mouth, the motorized wheels may also be reversed and turned on and off so that it clears obstructions when required. Linear solenoid actuators will also serve to clear obstructions by bottles. These solenoids can be placed on the sides of the exit mouth where bottle obstructions are most likely to jam the entire mechanism. Solenoids were prototyped by perturbing the obstructions with a pencil. Motorized wheels were prototyped through manually moving a high-friction track underneath the bottles. The prototype solenoid agitators appeared to be much more reliable than the motorized wheels. In addition, motorized wheels will require continuous power to function as opposed to solenoids, which can be operated much more simply.

Based on reliability and simplicity, linear solenoid actuators to agitate the bottles were chosen as the jam prevention mechanism.

With only the oscillation of the V-shaped cradle, it is possible for 2 or 3 bottles to fall in a single cycle as a tightly-packed queue. The identification mechanism requires bottles to be entered one at a time and dispensed one at a time. In order to accommodate this, a separation mechanism is required to enter the bottles one-by-one after the bottle exits the cradle. The initial design concept for a separation mechanism involves a system of two conveyor belts and a single sensor: bottles will exit the cradle and queue on the first conveyor belt, but eventually half of a bottle will spill onto the second conveyor belt. A sensor will detect this and stop the first conveyor belt while continuing the second conveyor belt. When the bottle leaves the sights of the sensor, the first conveyor belt is resumed. This system results in a relatively uniform and separate distribution of bottles.

Another design concept for separation that was conceived was a sensor located just past the exit of the cradle. Normally, the cradle rocks back and forth, but when a bottle falls out of the mouth, the sensor detects it and the cradle is forced back with its mouth pointing upward. This will allow for the bottle that triggered the sensor to exit the cradle and fall into the identification mechanism but prevents any additional bottles from following. The cradle would remain in this position until the first bottle is sorted, after which the rocking will resume. The controlled rocking movement will not introduce a lot of additional complexity to the design as opposed to the conveyor belt system. Controlled rocking as a means of separation takes up less space and does not add much weight or cost to the design.

Based on the principle of simplicity, as well as objectives pertaining to space, weight, and cost, a controlled rocking mechanism as a means of bottle separation is preferred over a conveyor belt system.

Decision Making for Identification Mechanism

Firstly, identifying bottles using sensors was preferred over mechanical methods to decrease the number of moving parts and actuations in the design. This is to avoid possible mechanical failures which can affect the sorting process. A sensor configuration that is smart and uses the fewest number of sensors as possible is needed to minimize the complexity of the machine. The current sensor configuration in Figure 22 cuts down the number of sensors needed in the preliminary design (Figure 23) significantly, thus simplifying the identification process. This results in a faster and more reliable identification mechanism.

Decision Making for Delivery Mechanism

The design concept involving moving bins has the advantage of not having the identification mechanism move. Due to the large amount of circuitry present on the identification mechanism as well as concerns regarding alignment between various sorts, moving the identification chamber introduces multiple problems. However, the concern with the rotating bins design is the capability of the motor which rotates all four bins. Although the bins will not be very heavy, the radius of the over system might present some complications that can be avoided in other designs.

By having an intermediate chute, it will require about the same amount of the circuitry as the rotating bins design, since both systems interpret the results from the identification mechanism and translates it into servo motor movements. However, since the intermediate chute involved rotation of a much smaller object, weight and cost of the design is improved.

APPENDIX C: CALCULATIONS

Power consumption calculations with factor of safety:

Power req:

3x SG90 Servos: 5V, 650 mA (stall) → 3.25W

↳ addicore.com

1x FS90R Cts Servo: 5V, 550mA (stall) → 2.75W

↳ datasheet.

2x TCRT 5000: 200 mW

1x LTR-3208e: 100 mW

1x LTE-5208a: 150 mW.

1x Laser: 5V, $I = \frac{5}{330}$ A. → 75.8 mW.

1x Laser detector: (assume same as IR phototransistor), 100 mW.

1x DC Gear Motor: 6V, 3A (stall) → 18 W.

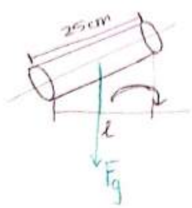
Total: ~ 28.3W.

36 W power supply OK.

FoS = 36/28.3 = 1.27

Determining motor required by calculating torque:

Motor for Shute for Delivery Arm:



$\tau = F \cdot r = I\alpha$

Assume mass of the shute:
$m \approx 250g$

$\tau = (0.250\,kg)(9.81\,m/s^2)\left(\frac{0.177m}{2}\right)$
$= 0.22\,Nm$

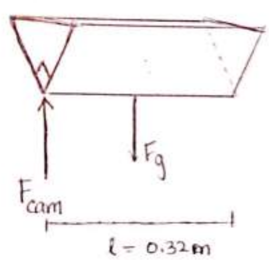
$l = \frac{0.25m}{\sqrt{2}} = 0.177m$

∴ The moto must have a torque greater than 0.22 Nm

$\tau > 0.22Nm$

Motor attached to Cam Mechanism to Actuate V-Shaped Cradle

Torque: $\tau = F \cdot r = I\alpha$



In order to displace the ~~ramp~~ cradle:
$F_{cam} > F_g$

Maximum mass of cradle:
$= 10(m_{bottle}) + m_{cradle}$
$= 10(0.025g) + (0.100g)$
$= 0.350\,kg$

$\tau = (0.350\,kg)(9.81\,m/s^2)\left(\frac{0.32m}{2}\right)$
$= 0.55\,Nm$

∴ The motor must have a torque greater than 0.55 Nm

$\tau > 0.55\,Nm$

APPENDIX D: CODE

D-1 main.c

```
/*
 * File:   main.c
 * Author: Maliha Islam
 * Description: Complete code for a bottle sorting machine that sorts four types
 *              bottles: YOP with/without cap, ESKA with/ without cap;
 *              machine interface and operation code
 * Reference Sources:
 *  DC PWM: https://electrosome.com/dc-motor-l293d-pic-microcontroller/
 *  LCD + Keypad Interface: Sample code provided by Instructor
 *  Servo PWM: https://electrosome.com/servo-motor-pic-microcontroller/
 *  RTC: Sample code provided by Instructor
 *  EEPROM: http://stackoverflow.com/questions/38436675/pic18f2520-mplab-x-xc8-eeprom
 * Created on February 19, 2017, 12:11 PM
 */


// <editor-fold defaultstate="collapsed" desc=" VARIABLES, CONSTANTS AND DECLARATIONS ">
#include <xc.h>
#include <stdio.h>
#include <stdbool.h>
#include <math.h>
#include <stdint.h>
#include "configBits.h"
#include "constants.h"
#include "lcd.h"
#include "I2C.h"
#include "macros.h"

//void set_time(void);
void delay(int);
void standby_mode(unsigned char[]);
void update_time(unsigned char[]);
void print(char[], char[]);
```

```
void operation(void);
int time_diff(unsigned char[], unsigned char[]);
void elapsed_time(void);
int dec2hex(int);
void wait(void);
void key_release(void);
void opentoplid(void);
void closetoplid(void);
void openbotlid(void);
void closebotlid(void);
void readADC(char channel);
void lcdclear(void);
uint8_t read_eeprom(uint16_t address);
void write_eeprom(uint16_t address, uint8_t data);

const char keys[] = "123A456B789C*0#D";
const char curr_time[7] =    {0x00, //50 Seconds
                        0x47, //13 Minutes
                        0x00, //24 hour mode, set to 21:00
                        0x02, //Wednesday
                        0x11, //22nd
                        0x04, //February
                        0x17};//2017

unsigned char time[7];
unsigned char start_time[7];
unsigned char end_time[7];
int flag, flag1, flag2, flag3, flag4, flag5;
int yop_a = 0, yop_b = 0, eska_a = 0, eska_b = 0;
int totalbot = 0;
int run = 0;
unsigned char pressed;

//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" BOTTLE CATEGORIES ">
enum category {
```

```
    YOP,
    ESKA,
    CAP,
    NOCAP
};
enum category brand;
enum category type;

//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" STATES ">
enum state {
    STANDBY,
    OPERATION,
    END,
    LOG
};
enum state curr_state = STANDBY;
//</editor-fold>

void main(void) {

    // <editor-fold defaultstate="collapsed" desc=" STARTUP SEQUENCE ">

    TRISA = 0b11011110; // Set Port A as all input, except RA5 (top lid) and RA0 (test LED)
    TRISB = 0b11111010; // Set Port B as all input, except RB0 and RB2 (DC motor)
    TRISC = 0b01011001; // Set Port C as all output, except RC3 and RC4 (I2C), RC0 (laser detector), RC6(IR
detector)
    TRISD = 0x00; // Set Port D as all output
    TRISE = 0x00; // Set Port E as all output

    LATA = 0x00;
    LATB = 0x00;
    LATC = 0x00;
    LATD = 0x00;
    LATE = 0x00;
```

```
ADCON0 = 0x00;  //Disable ADC
ADCON1 = 0x0B;  //AN0 to AN3 used as analog input
CVRCON = 0x00; // Disable CCP reference voltage output
CMCONbits.CIS = 0;
ADFM = 1;

RB0 = 1;        //RB0 is motor IN1
RB2 = 1;        //RB2 is motor IN2, both high stops motor

nRBPU = 0;

I2C_Master_Init(10000); //Initialize I2C Master with 100KHz clock

INT1IE = 1;
ei(); // Enable all interrupts

initLCD();
// <editor-fold defaultstate="collapsed" desc=" WRITING TO EEPROM ">

        write_eeprom(0x30,0x0);
        write_eeprom(0x31,0x0);
        write_eeprom(0x32,0x0);
        write_eeprom(0x33,0x0);


        write_eeprom(0x20,0x0);
        write_eeprom(0x21,0x0);
        write_eeprom(0x22,0x0);
        write_eeprom(0x23,0x0);


        write_eeprom(0x10,0x0);
        write_eeprom(0x11,0x0);
        write_eeprom(0x12,0x0);
        write_eeprom(0x13,0x0);
```

```
                write_eeprom(0x00,0x0);
                write_eeprom(0x01,0x0);
                write_eeprom(0x02,0x0);
                write_eeprom(0x03,0x0);

//</editor-fold>
//</editor-fold>

while (1){
    //update_time(time);
    //set_time();

    switch(curr_state){
        case STANDBY:
            flag = 0, flag1 = 0, flag2 = 0, flag3 = 0, flag4 = 0, flag5 = 0;
            lcdclear();
            __lcd_home();
            while(keys[(PORTB & 0xF0)>>4] != '0'){
                // RB1 is the interrupt pin, so if there is no key pressed, RB1 will be 0
                // the PIC will be in standby mode until a key press is signaled
                standby_mode(time);                    //SCREEN 1: Time and Date
            }
            key_release();

            while (!flag){
                lcdclear();
                print("*: Start     |<C","1: Past Logs |D>");          //SCREEN 2: Menu Part 1 (Start + Past Logs)
                wait();
                key_release();

                if (pressed == '*'){
                    // <editor-fold defaultstate="collapsed" desc=" OPERATION START SCREEN ">
                    yop_a = 0;
                    yop_b = 0;
                    eska_a = 0;
                    eska_b = 0;
                    totalbot = 0;
```

```
            if (run == 5){
                run = 1;
            }
            else{
                run++;
            }

            lcdclear();
            print("    Sorting    ","    has begun!   ");              //SCREEN 2A: Starting Operation
            delay(2);

            lcdclear();
            print("Sorting...        ","Press # to STOP    ");
            update_time(start_time);
            update_time(end_time);
            curr_state = OPERATION;
            //</editor-fold>
            break;
        }
        else if (pressed == '1'){
            flag1=0;
            while(!flag1){
                lcdclear();
                print("4: Run 1     |<C","5: Run 2     |D>");           //SCREEN 2B1: Past Logs Part 1 (Run 1 + Run
2)
                wait();
                key_release();
                if (pressed == '4'){
                    // <editor-fold defaultstate="collapsed" desc=" RUN 1 SCREEN ">
                    lcdclear();
                    printf("YOP A: %i B: %i|<C",(int)read_eeprom(0x30),(int)read_eeprom(0x31));   //SCREEN
2B1i: Run 1 Bottle Count
                    __lcd_newline();
                    printf("ESK A: %i B: %i   ",(int)read_eeprom(0x32),(int)read_eeprom(0x33));
                    wait();
                    key_release();
                    //</editor-fold>
```

```
                }
                else if (pressed == '5'){
                    // <editor-fold defaultstate="collapsed" desc=" RUN 2 SCREEN ">
                    lcdclear();
                    printf("YOP A: %i B: %i|<C",read_eeprom(0x20),read_eeprom(0x21));    //SCREEN 2B1ii: Run
2 Bottle Count
                    __lcd_newline();
                    printf("ESK A: %i B: %i    ",read_eeprom(0x22),read_eeprom(0x23));
                    wait();
                    key_release();
                    //</editor-fold>
                }
                else if (pressed == 'D'){
                    flag2= 0;
                    while(!flag2){
                        lcdclear();
                        print("6: Run 3      |<C","7: Run 4          ");           //SCREEN 2B2: Past Logs Part 2 (Run 3 +
Run 4)
                        wait();
                        key_release();
                        if (pressed == '6'){
                            // <editor-fold defaultstate="collapsed" desc=" RUN 3 SCREEN ">
                            lcdclear();
                            printf("YOP A: %i B: %i|<C",(int)read_eeprom(0x10),(int)read_eeprom(0x11));    //SCREEN
2B2i: Run 3 Bottle Count
                            __lcd_newline();
                            printf("ESK A: %i B: %i    ",(int)read_eeprom(0x12),(int)read_eeprom(0x13));
                            wait();
                            key_release();
                            //</editor-fold>
                        }
                        else if (pressed == '7'){
                            // <editor-fold defaultstate="collapsed" desc=" RUN 4 SCREEN ">
                            lcdclear();
                            printf("YOP A: %i B: %i|<C",(int)read_eeprom(0x00),(int)read_eeprom(0x01));    //SCREEN
2B2ii: Run 4 Bottle Count
                            __lcd_newline();
```

```
                    printf("ESK A: %i B: %i   ",(int)read_eeprom(0x02),(int)read_eeprom(0x03));
                    wait();
                    key_release();
                    //</editor-fold>
                }
                else if (pressed == 'C'){
                    flag2 = 1;
                }
            }
        }
        else if (pressed == 'C'){
            flag1 = 1;
        }
    }
}
else if (pressed == 'C'){
    curr_state = STANDBY;
    break;
}
else if (pressed == 'D'){
    flag3=0;
    while (!flag3){
        lcdclear();
        print("2: About     |<C","3: Credits         ");          //SCREEN 3: MENU PART 2 (About + Credits)
        wait();
        key_release();
        if (pressed == '2'){
            // <editor-fold defaultstate="collapsed" desc=" ABOUT SCREEN ">
            flag4=0;
            while (!flag4){
                lcdclear();
                print("EcoSort      |<C","Version: v1.0|D>");          //SCREEN 3A1: About Part 1
                wait();
                key_release();
                if (pressed == 'C'){
                    flag4 = 1;
                }
```

```
            else if (pressed == 'D'){
                lcdclear();
                print("Copyright(C)2017","InnovaTech   |<C");        //SCREEN 3A2: About Part 2
                wait();
                key_release();

            }
        }
        //</editor-fold>
    }
    else if (pressed == '3'){
        // <editor-fold defaultstate="collapsed" desc=" CREDITS SCREEN ">
        flag4=0;
        while(!flag4){
            lcdclear();
            print("Mechanics:   |<C","R. Jose      |D>");          //SCREEN 3B1: Mechanics Credits
            wait();
            key_release();
            if (pressed == 'C'){
                flag4 = 1;
            }
            else if (pressed == 'D'){
                flag5=0;
                while(!flag5){
                    lcdclear();
                    print("Circuitry:   |<C","J. Qiu       |D>");   //SCREEN 3B2: Circuitry Credits
                    wait();
                    key_release();
                    if (pressed == 'D'){
                        lcdclear();
                        print("Programming: |<C","M. Islam        ");        //SCREEN 3B3: Programming
Credits

                        wait();
                        key_release();
                    }
                    else if (pressed == 'C'){
                        flag5=1;
```

```
                        }
                    }
                }
            }
            //</editor-fold>
        }
        else if (pressed == 'C'){
            flag3 = 1;
        }
    }
}
}

case OPERATION:
    //update_time(start_time);

    while ((time_diff(start_time,end_time)<180 || totalbot < 10) && curr_state == OPERATION){
        if (curr_state == OPERATION){
            closebotlid();
            opentoplid();
        }
        readADC(3); //reads analog input from RA3 (IR TCRT5000L) on delivery arm

        while ((ADRESH*256 + ADRESL > 0x19))
        {
            readADC(3);
            RC7 = 1;
            __delay_us(2000);
            RC7 = 0;
            __delay_us(18000);
        }

        unsigned int counter = 0;
        while (RC0 && curr_state==OPERATION){

            //wait here until laser detector (RC0) detects a bottle in the
            //testing chamber; 1 = no bottle, 0 = bottle
```

```
        RB0 = 0;        // motor start
        for (unsigned int i = 0; i<31; i++){    //PWM for motor
            PORTEbits.RE0 = 1;
            __delay_us(18300);
            PORTEbits.RE0 = 0;
            __delay_us(1700);

        }
        RB0 = 1;                //motor stop
        __delay_ms(3000);
        counter++;
        if ((counter == 5) && RC0){     //foolproof for Eska not detected by the laser beam
            openbotlid();
            delay(1);
            closebotlid();
        }
        if ((yop_a>=1) && (yop_b>=1) && (eska_a>=1) && (eska_b>=1) && (counter==10)){ //loading bin
stops moving after 10 cycles if at least 4 bottles has been sorted
            update_time(end_time);
            curr_state = END;
            counter = 0;
            break;
        }


    }

    if (curr_state == OPERATION){
        closetoplid();
    }

    // <editor-fold defaultstate="collapsed" desc=" IDENTIFICATION ALGORITHM ">

    readADC(2); //reads analog input from RA2 (IR TCRT5000) on testing chamber

    if (RC6){ //RC6 is IR detector, input 1 for opaque, 0 for clear
```

```
    brand = YOP;
}
else if (!RC6){
    brand = ESKA;
}

if (brand == YOP && RA4 == 0 && (ADRESL < 0x14)){
    //RA4 is IR TCRT5000 module, RA2 is IR TCRT5000
    type = CAP;
}
else if (brand == ESKA && (RA4 == 0 || (ADRESL < 0x14))){
    type = CAP;
}
else{
    type = NOCAP;
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" DELIVERY CHUTE ROTATION ">

//RC7 is the pin for continuous servo


if (brand == ESKA && type == CAP && curr_state == OPERATION){
    openbotlid();

    eska_a++;                          //add to ESKA A count
}
else if (brand == YOP && type == CAP && curr_state == OPERATION){
// <editor-fold defaultstate="collapsed" desc=" GO TO 90 DEGREES ">
    unsigned int i;
    for(i=0;i<5;i++)                   //rotate chute to 90 degrees
    {

      RC7 = 1;
      __delay_us(2300);
      RC7 = 0;
```

```
      __delay_us(17700);
    }
    openbotlid();

//</editor-fold>
    yop_a++;                    //add to YOP A count
}

else if (brand == YOP && type == NOCAP && curr_state == OPERATION){
// <editor-fold defaultstate="collapsed" desc=" GO TO 180 DEGREES ">
    unsigned int i;
    for(i=0;i<13;i++)           //rotate chute to 180 degrees
    {

      RC7 = 1;
      __delay_us(2300);
      RC7 = 0;
      __delay_us(17700);
    }
    openbotlid();

//</editor-fold>
    yop_b++;                    //add to YOP B count
}

else if (brand == ESKA && type == NOCAP && curr_state == OPERATION){
// <editor-fold defaultstate="collapsed" desc=" GO TO 270 DEGREES ">
    unsigned int i;
    for(i=0;i<21;i++)           //rotate chute to 270 degrees
    {

      RC7 = 1;
      __delay_us(2300);
      RC7 = 0;
      __delay_us(17700);
    }
    openbotlid();
```

```
            //</editor-fold>
                eska_b++;                    //add to ESKA B count
            }
            //</editor-fold>

            totalbot++;
            if (totalbot == 10){
                update_time(end_time);
                curr_state = END;
                break;
            }

            update_time(end_time);
        }

        if (curr_state != OPERATION){
            update_time(end_time);
            break;
        }
        update_time(end_time);
        curr_state = END;
        break;

    case END:
        //update_time(end_time);
        RB0=1; //stop motor
        RC7=0; //stop cts servo
        opentoplid();
        closebotlid();

        lcdclear();
        print(" Operation has  ","   terminated   ");
        delay(2);
        RA0 = 1; //DEBUG
        curr_state = LOG;
        break;
```

```
case LOG:
    flag = 0, flag2 = 0;

    // <editor-fold defaultstate="collapsed" desc=" WRITING TO EEPROM ">
    if (run == 1){
        write_eeprom(0x30,(uint8_t)yop_a);
        write_eeprom(0x31,(uint8_t)yop_b);
        write_eeprom(0x32,(uint8_t)eska_a);
        write_eeprom(0x33,(uint8_t)eska_b);
    }
    else if (run == 2){
        write_eeprom(0x20,(uint8_t)yop_a);
        write_eeprom(0x21,(uint8_t)yop_b);
        write_eeprom(0x22,(uint8_t)eska_a);
        write_eeprom(0x23,(uint8_t)eska_b);
    }
    else if (run == 3){
        write_eeprom(0x10,(uint8_t)yop_a);
        write_eeprom(0x11,(uint8_t)yop_b);
        write_eeprom(0x12,(uint8_t)eska_a);
        write_eeprom(0x13,(uint8_t)eska_b);
    }
    else if (run == 4) {
        write_eeprom(0x00,(uint8_t)yop_a);
        write_eeprom(0x01,(uint8_t)yop_b);
        write_eeprom(0x02,(uint8_t)eska_a);
        write_eeprom(0x03,(uint8_t)eska_b);
    }
    //</editor-fold>

    while(!flag && curr_state == LOG){
        lcdclear();
        print("  Collect Log?  ", "< B Home|Yes A >");

        wait();
        key_release();
```

```
                if (pressed == 'A'){
                    flag2=0;
                    while (!flag2){
                        lcdclear();
                        printf("YOP A: %i B: %i|<C",yop_a,yop_b);
                        __lcd_newline();
                        printf("ESK A: %i B: %i|D>",eska_a, eska_b);
                        wait();
                        key_release();
                        if (pressed == 'D'){
                            totalbot = eska_a + eska_b + yop_a + yop_b;
                            lcdclear();
                            printf("Total: %i    |<C",totalbot);
                            __lcd_newline();
                            printf("Time: %is       ",time_diff(start_time,end_time)-2);
                            wait();
                            key_release();
                        }
                        else if (pressed == 'C'){
                            flag2 = 1;
                        }
                    }
                }
            }
            break;
        }
    }

    return;
}


// <editor-fold defaultstate="collapsed" desc=" lcdclear FUNCTION ">
void lcdclear(void){
    __lcd_clear();
    __delay_ms(2);
}
```

```
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" set_time FUNCTION ">
void set_time(void){
    I2C_Master_Start(); //Start condition
    I2C_Master_Write(0b11010000); //7 bit RTC address + Write
    I2C_Master_Write(0x00); //Set memory pointer to seconds
    for(char i=0; i<7; i++){
        I2C_Master_Write(curr_time[i]);
    }
    I2C_Master_Stop(); //Stop condition
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" standby_mode FUNCTION ">
void standby_mode(unsigned char time[]){
    update_time(time);
    di();
    printf("%02x/%02x/%02x MENU:", time[6],time[5],time[4]);    //Print date in YY/MM/DD
    __lcd_newline();
    printf("%02x:%02x:%02x Press 0", time[2],time[1],time[0]);    //HH:MM:SS
    ei();
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" update_time FUNCTION ">
void update_time(unsigned char time[]){
    //Reset RTC memory pointer
    I2C_Master_Start(); //Start condition
    I2C_Master_Write(0b11010000); //7 bit RTC address + Write
    I2C_Master_Write(0x00); //Set memory pointer to seconds
    I2C_Master_Stop(); //Stop condition

    //Read Current Time
    I2C_Master_Start();
    I2C_Master_Write(0b11010001); //7 bit RTC address + Read
    for(unsigned char i=0;i<0x06;i++){
```

```
        time[i] = I2C_Master_Read(1);
    }
    time[6] = I2C_Master_Read(0);      //Final Read without ack
    I2C_Master_Stop();
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" time_diff FUNCTION ">
int time_diff(unsigned char time1[], unsigned char time2[]) {
    int hr1, hr2, min1, min2, s1, s2;
    int d1, d2, d3;
    hr1 = time1[2]; hr2 = time2[2]; min1 = time1[1]; min2 = time2[1];
    s1 = time1[0]; s2 = time2[0];

    d1 = dec2hex(hr2) - dec2hex(hr1);
    d2 = dec2hex(min2) - dec2hex(min1);
    d3 = dec2hex(s2) - dec2hex(s1);

    return 3600*d1 + 60*d2 + d3;
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" delay FUNCTION ">
void delay(int seconds) {
    for (int i = 0; i <= seconds; i ++) {
        __delay_1s();
    }
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" dec2hex FUNCTION ">
int dec2hex(int num) {
    int i = 0, quotient = num, temp, hexnum = 0;


    while (quotient != 0) {
        temp = quotient % 16;
```

```
        hexnum += temp*pow(10, i);

        quotient = quotient / 16;
        i += 1;
    }
    return hexnum;
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" wait FUNCTION ">
void wait(void){
    while(PORTBbits.RB1 == 0){
        // Wait until a key is pressed
    }
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" key_release FUNCTION ">
void key_release(void){

    while(PORTBbits.RB1 == 1){
        // Wait until the key has been released
        pressed = keys[(PORTB & 0xF0)>>4];
    }
    Nop();  //Apply breakpoint here because of compiler optimizations
    Nop();
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" print FUNCTION ">
void print(char line1[], char line2[]) {
    if (line2 == ""){
        printf(line1);
    }
    else{
        printf(line1);
```

```
        __lcd_newline();
        printf(line2);
    }
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" readADC FUNCTION ">
void readADC(char channel){
    // Select A2D channel to read
    ADCON0 = ((channel <<2));
    ADON = 1;
    ADCON0bits.GO = 1;
   while(ADCON0bits.GO_NOT_DONE){__delay_ms(5);}
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" KEYPAD INTERRUPTS ">
void interrupt keypressed(void) {
    if(INT1IF){
        switch(keys[(PORTB & 0xF0) >> 4]){
            case '#':
                update_time(end_time);
                RC7=0; //stop cts servo
                RB0=1; //stop motor
                closebotlid();
                opentoplid();
                lcdclear();
                print(" Operation has  ","   terminated   ");
                delay(1);
                RA0 = 1;
                curr_state = LOG;
                break;

            case 'B':
                curr_state = STANDBY;
                break;
```

```
            default:
                break;
        }
        INT1IF = 0;     //Clear flag bit
    }
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" 180 SERVO FUNCTIONS ">
void opentoplid(void)
{
    unsigned int i;
        for(i=0;i<50;i++)
        {
          PORTAbits.RA5 = 1;
          __delay_us(1200);
          PORTAbits.RA5 = 0;
          __delay_us(18800);
        }
}

void closetoplid(void)
{
    unsigned int i;
        for(i=0;i<50;i++)
        {
          PORTAbits.RA5 = 1;
          __delay_us(2700);
          PORTAbits.RA5 = 0;
          __delay_us(17300);
        }
}

void openbotlid(void)
{
  unsigned int i;
  for(i=0;i<50;i++)
```

```
  {
    RC5 = 1;
    __delay_us(1250);
    RC5 = 0;
    __delay_us(18750);
  }
}

void closebotlid(void)
{
  unsigned int i;
  for(i=0;i<50;i++)
  {
    RC5 = 1;
    __delay_us(2700);
    RC5 = 0;
    __delay_us(17300);
  }
}
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc=" EEPROM R/W FUNCTIONS ">
uint8_t read_eeprom(uint16_t address)
{

    // Set address registers
    EEADRH = (uint8_t)(address >> 8);
    EEADR = (uint8_t)address;

    EECON1bits.EEPGD = 0;       // Select EEPROM Data Memory
    EECON1bits.CFGS = 0;        // Access flash/EEPROM NOT config. registers
    EECON1bits.RD = 1;          // Start a read cycle

    // A read should only take one cycle, and then the hardware will clear
    // the RD bit
    while(EECON1bits.RD == 1);
```

```
    return EEDATA;              // Return data

}

void write_eeprom(uint16_t address, uint8_t data)
{
    // Set address registers
    EEADRH = (uint8_t)(address >> 8);
    EEADR = (uint8_t)address;

    EEDATA = data;          // Write data we want to write to SFR
    EECON1bits.EEPGD = 0;   // Select EEPROM data memory
    EECON1bits.CFGS = 0;    // Access flash/EEPROM NOT config. registers
    EECON1bits.WREN = 1;    // Enable writing of EEPROM (this is disabled again after the write completes)

    // The next three lines of code perform the required operations to
    // initiate a EEPROM write
    EECON2 = 0x55;          // Part of required sequence for write to internal EEPROM
    EECON2 = 0xAA;          // Part of required sequence for write to internal EEPROM
    EECON1bits.WR = 1;      // Part of required sequence for write to internal EEPROM

    // Loop until write operation is complete
    while(PIR2bits.EEIF == 0)
    {
        continue;   // Do nothing, are just waiting
    }

    PIR2bits.EEIF = 0;      //Clearing EEIF bit (this MUST be cleared in software after each write)
    EECON1bits.WREN = 0;    // Disable write (for safety, it is re-enabled next time a EEPROM write is performed)
}
//</editor-fold>
```

D-2 lcd.c

```
/*
 * File:   lcd.c
 * Author: True Administrator (Provided by Instructor)
```

```
 *
 * Created on July 18, 2016, 12:11 PM
 */

#include <xc.h>
#include "configBits.h"
#include <stdio.h>
#include "lcd.h"
#include "constants.h"

void initLCD(void) {
    __delay_ms(15);
    //lcdInst(0b00110011);        //Force into 8bit mode
    lcdInst(0b00110011);        //Should require only three commands, but
    lcdInst(0b00110010);        //Seems to be demanding five in this case.
    lcdInst(0b00101000);
    lcdInst(0b00001111);
    lcdInst(0b00000110);
    lcdInst(0b00000001);
    __delay_ms(15);
}

void lcdInst(char data) {
    RS = 0;
    lcdNibble(data);
}

void putch(char data){
    RS = 1;
    lcdNibble(data);
}

void lcdNibble(char data){
    // Send of 4 most sig bits, then the 4 least sig bits (MSD,LSD)
    char temp = data & 0xF0;
    LATD = LATD & 0x0F;
    LATD = temp | LATD;
```

```
    E = 0;
    __delay_us(LCD_DELAY);
    E = 1;
    __delay_us(LCD_DELAY);

    data = data << 4;

    temp = data & 0xF0;
    LATD = LATD & 0x0F;
    LATD = temp | LATD;

    E = 0;
    __delay_us(LCD_DELAY);
    E = 1;
    __delay_us(LCD_DELAY);
}
```

D-3 I2C.c

```
/*
 * File:   I2C.c
 * Author: Administrator (Provided by Instructor)
 *
 * Created on August 4, 2016, 3:22 PM
 */


#include <xc.h>
#include "I2C.h"
#include "configBits.h"

void I2C_Master_Init(const unsigned long c)
{
  // See Datasheet pg171, I2C mode configuration
  SSPSTAT = 0b00000000;
  SSPCON1 = 0b00101000;
```

```
  SSPCON2 = 0b00000000;
  SSPADD = (_XTAL_FREQ/(4*c))-1;
  TRISC3 = 1;        //Setting as input as given in datasheet
  TRISC4 = 1;        //Setting as input as given in datasheet
}

void I2C_Master_Wait()
{
  while ((SSPSTAT & 0x04) || (SSPCON2 & 0x1F));
}

void I2C_Master_Start()
{
  I2C_Master_Wait();
  SEN = 1;
}

void I2C_Master_RepeatedStart()
{
  I2C_Master_Wait();
  RSEN = 1;
}

void I2C_Master_Stop()
{
  I2C_Master_Wait();
  PEN = 1;
}

void I2C_Master_Write(unsigned d)
{
  I2C_Master_Wait();
  SSPBUF = d;
}

unsigned char I2C_Master_Read(unsigned char a)
{
```

```
  unsigned char temp;
  I2C_Master_Wait();
  RCEN = 1;
  I2C_Master_Wait();
  temp = SSPBUF;
  I2C_Master_Wait();
  ACKDT = (a)?0:1;
  ACKEN = 1;
  return temp;
}

void delay_10ms(unsigned char n) {
    while (n-- != 0) {
        __delay_ms(5);
    }
}
```

D-4 configBits.h

```
/*
 * PIC18F4620 Configuration Bits
 * To generate this configuration, Window->PIC Memory Views->Configuration Bits
 * and select from the options provided. You can then generate the source and
 * paste it into a header file like this one.
 */

// CONFIG1H
#pragma config OSC = HS         // Oscillator Selection bits (HS oscillator)
#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF       // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF       // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = SBORDIS  // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only
(SBOREN is disabled))
#pragma config BORV = 3         // Brown Out Reset Voltage bits (Minimum setting)
```

```
// CONFIG2H
#pragma config WDT = OFF        // Watchdog Timer Enable bit (WDT disabled (control is placed on the SWDTEN bit))
#pragma config WDTPS = 32768    // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = PORTC   // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON      // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
#pragma config LPT1OSC = OFF    // Low-Power Timer1 Oscillator Enable bit (Timer1 configured for higher power operation)
#pragma config MCLRE = ON       // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON      // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = OFF        // Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
#pragma config XINST = OFF      // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF        // Code Protection bit (Block 0 (000800-003FFFh) not code-protected)
#pragma config CP1 = OFF        // Code Protection bit (Block 1 (004000-007FFFh) not code-protected)
#pragma config CP2 = OFF        // Code Protection bit (Block 2 (008000-00BFFFh) not code-protected)
#pragma config CP3 = OFF        // Code Protection bit (Block 3 (00C000-00FFFFh) not code-protected)

// CONFIG5H
#pragma config CPB = OFF        // Boot Block Code Protection bit (Boot block (000000-0007FFh) not code-protected)
#pragma config CPD = OFF        // Data EEPROM Code Protection bit (Data EEPROM not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF       // Write Protection bit (Block 0 (000800-003FFFh) not write-protected)
#pragma config WRT1 = OFF       // Write Protection bit (Block 1 (004000-007FFFh) not write-protected)
#pragma config WRT2 = OFF       // Write Protection bit (Block 2 (008000-00BFFFh) not write-protected)
#pragma config WRT3 = OFF       // Write Protection bit (Block 3 (00C000-00FFFFh) not write-protected)
```

```
// CONFIG6H
#pragma config WRTC = OFF       // Configuration Register Write Protection bit (Configuration registers (300000-3000FFh) not write-protected)
#pragma config WRTB = OFF       // Boot Block Write Protection bit (Boot Block (000000-0007FFh) not write-protected)
#pragma config WRTD = OFF       // Data EEPROM Write Protection bit (Data EEPROM not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF      // Table Read Protection bit (Block 0 (000800-003FFFh) not protected from table reads executed in other blocks)
#pragma config EBTR1 = OFF      // Table Read Protection bit (Block 1 (004000-007FFFh) not protected from table reads executed in other blocks)
#pragma config EBTR2 = OFF      // Table Read Protection bit (Block 2 (008000-00BFFFh) not protected from table reads executed in other blocks)
#pragma config EBTR3 = OFF      // Table Read Protection bit (Block 3 (00C000-00FFFFh) not protected from table reads executed in other blocks)

// CONFIG7H
#pragma config EBTRB = OFF      // Boot Block Table Read Protection bit (Boot Block (000000-0007FFh) not protected from table reads executed in other blocks)

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#include <xc.h>
#include <string.h>

#define _XTAL_FREQ 8000000      // Define osc freq for use in delay macros
```

D-5 constants.h

```
/*
 * File:   constants.h
 * Author: Administrator, Maliha Islam
 *
 * Created on August 11, 2016, 2:41 PM
 */
```

```
#ifndef CONSTANTS_H
#define  CONSTANTS_H        //Prevent multiple inclusion

//LCD Control Registers
#define RS          LATDbits.LATD2
#define E           LATDbits.LATD3
#define  LCD_PORT    LATD   //On LATD[4,7] to be specific
#define LCD_DELAY   30


#endif    /* CONSTANTS_H */
```

D-6 I2C.h

```
void I2C_Master_Init(const unsigned long c);
void I2C_Master_Write(unsigned d);
unsigned char I2C_Master_Read(unsigned char a);
```

D-7 lcd.h

```
/*
 * File:   lcd.h
 * Author: Administrator
 *
 * Created on August 12, 2016, 4:24 PM
 */

#ifndef LCD_H
#define  LCD_H

void lcdInst(char data);
void lcdNibble(char data);
void initLCD(void);

#endif    /* LCD_H */
```

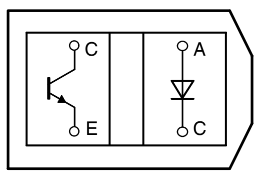
D-8 macros.h

```
/*
 * File:   macros.h
 * Author: Administrator
 *
 * Created on August 17, 2016, 2:42 PM
 */

#ifndef MACROS_H
#define  MACROS_H

#define __delay_1s() for(char i=0;i<100;i++){__delay_ms(10);}
#define __lcd_shift() lcdInst(0b00011111)
#define __lcd_newline() lcdInst(0b11000000);
#define __lcd_clear() lcdInst(0b00000001);
#define __lcd_home() lcdInst(0b10000000);
#define __bcd_to_num(num) (num & 0x0F) + ((num & 0xF0)>>4)*10


#endif  /* MACROS_H */
```

APPENDIX E: DATASHEETS

Reflective Optical Sensor with Transistor Output

19156_2



Top view 19156_1

FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 10.2 x 5.8 x 7
- Peak operating distance: 2.5 mm
- Operating range within > 20 % relative collector current: 0.2 mm to 15 mm
- Typical output current under test: I_C = 1 mA
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC

Pb-free
e2
RoHS
COMPLIANT

DESCRIPTION

The TCRT5000 and TCRT5000L are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light. The package includes two mounting clips. TCRT5000L is the long lead version.

APPLICATIONS

- Position sensor for shaft encoder
- Detection of reflective material such as paper, IBM cards, magnetic tapes etc.
- Limit switch for mechanical motions in VCR
- General purpose - wherever the space is limited

PRODUCT SUMMARY

PART NUMBER	DISTANCE FOR MAXIMUM CTR_{rel} (1) (mm)	DISTANCE RANGE FOR RELATIVE I_{out} > 20 % (mm)	TYPICAL OUTPUT CURRENT UNDER TEST (2) (mA)	DAYLIGHT BLOCKING FILTER INTEGRATED
TCRT5000	2.5	0.2 to 15	1	Yes
TCRT5000L	2.5	0.2 to 15	1	Yes

Notes

(1) CTR: current transfere ratio, I_{out}/I_{in}

(2) Conditions like in table basic charactristics/sensors

ORDERING INFORMATION

ORDERING CODE	PACKAGING	VOLUME (1)	REMARKS
TCRT5000	Tube	MOQ: 4500 pcs, 50 pcs/tube	3.5 mm lead length
TCRT5000L	Tube	MOQ: 2400 pcs, 48 pcs/tube	15 mm lead length

Note

(1) MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS (1)

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
INPUT (EMITTER)				
Reverse voltage		V_R	5	V
Forward current		I_F	60	mA
Forward surge current	$t_p \leq 10\ \mu s$	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25$ °C	P_V	100	mW
Junction temperature		T_j	100	°C

For technical questions, contact: sensorstechsupport@vishay.com

ABSOLUTE MAXIMUM RATINGS [1]				
PARAMETER	**TEST CONDITION**	**SYMBOL**	**VALUE**	**UNIT**
OUTPUT (DETECTOR)				
Collector emitter voltage		V_{CEO}	70	V
Emitter collector voltage		V_{ECO}	5	V
Collector current		I_C	100	mA
Power dissipation	$T_{amb} \leq 55$ °C	P_V	100	mW
Junction temperature		T_j	100	°C
SENSOR				
Total power dissipation	$T_{amb} \leq 25$ °C	P_{tot}	200	mW
Ambient temperature range		T_{amb}	- 25 to + 85	°C
Storage temperature range		T_{stg}	- 25 to + 100	°C
Soldering temperature	2 mm from case, t ≤ 10 s	T_{sd}	260	°C

Note
[1] T_{amb} = 25 °C, unless otherwise specified

ABSOLUTE MAXIMUM RATINGS

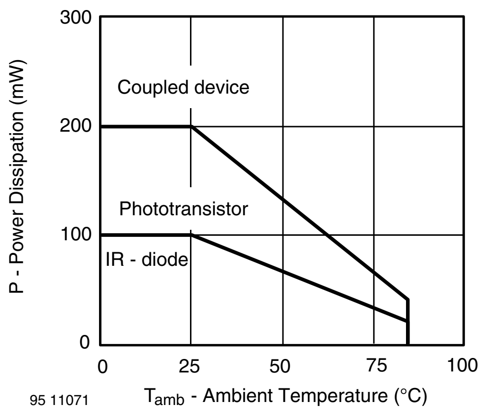


Fig. 1 - Power Dissipation Limit vs. Ambient Temperature

BASIC CHARACTERISTICS [1]						
PARAMETER	**TEST CONDITION**	**SYMBOL**	**MIN.**	**TYP.**	**MAX.**	**UNIT**
INPUT (EMITTER)						
Forward voltage	I_F = 60 mA	V_F		1.25	1.5	V
Junction capacitance	V_R = 0 V, f = 1 MHz	C_j		17		pF
Radiant intensity	I_F = 60 mA, t_p = 20 ms	I_e			21	mW/sr
Peak wavelength	I_F = 100 mA	λ_P	940			nm
Virtual source diameter	Method: 63 % encircled energy	d		2.1		mm
OUTPUT (DETECTOR)						
Collector emitter voltage	I_C = 1 mA	V_{CEO}	70			V
Emitter collector voltage	I_e = 100 µA	V_{ECO}	7			V
Collector dark current	V_{CE} = 20 V, I_F = 0 A, E = 0 lx	I_{CEO}		10	200	nA
SENSOR						
Collector current	V_{CE} = 5 V, I_F = 10 mA, D = 12 mm	I_C [2] [3]	0.5	1	2.1	mA
Collector emitter saturation voltage	I_F = 10 mA, I_C = 0.1 mA, D = 12 mm	V_{CEsat} [2] [3]			0.4	V

Note
[1] T_{amb} = 25 °C, unless otherwise specified
[2] See figure 3
[3] Test surface: mirror (Mfr. Spindler a. Hoyer, Part No. 340005)

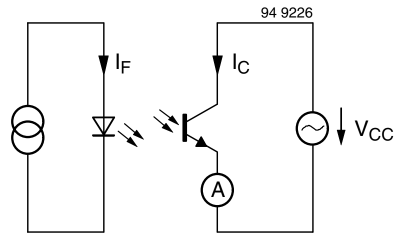


Fig. 2 - Test Circuit

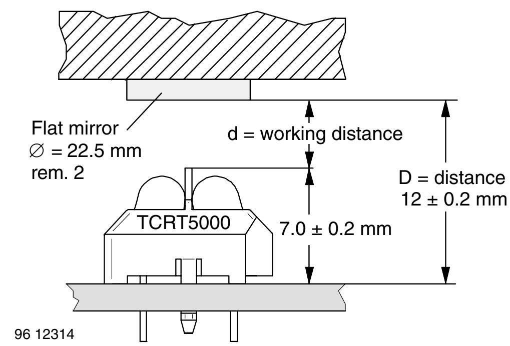


Fig. 3 - Test Circuit

BASIC CHARACTERISTICS

T_{amb} = 25 °C, unless otherwise specified

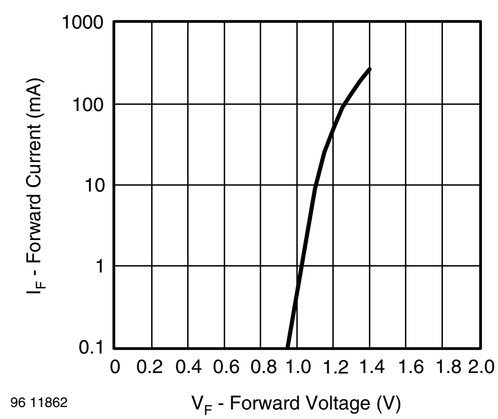


Fig. 4 - Forward Current vs. Forward Voltage

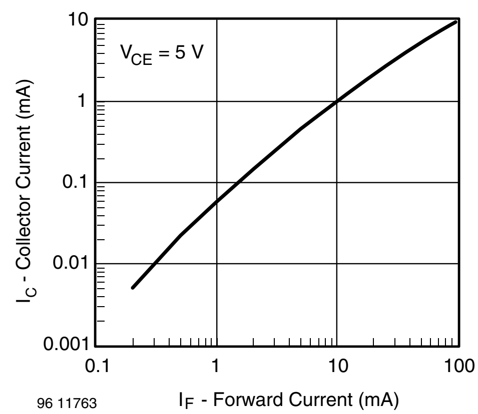


Fig. 6 - Collector Current vs. Forward Current

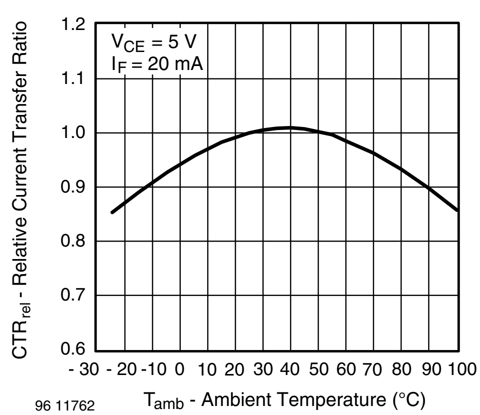


Fig. 5 - Relative Current Transfer Ratio vs. Ambient Temperature

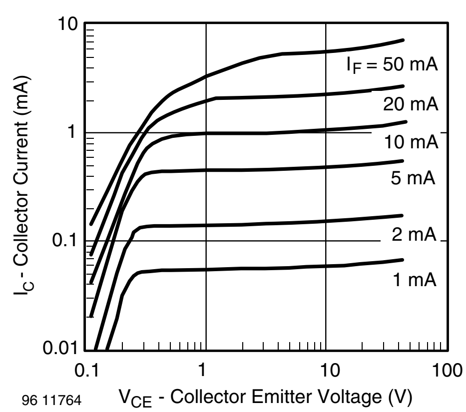


Fig. 7 - Collector Emitter Saturation Voltage vs. Collector Current

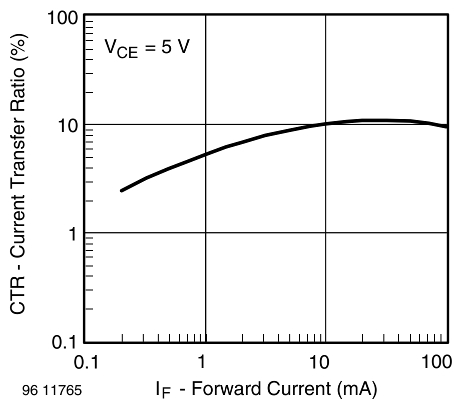


Fig. 8 - Current Transfer Ratio vs. Forward Current

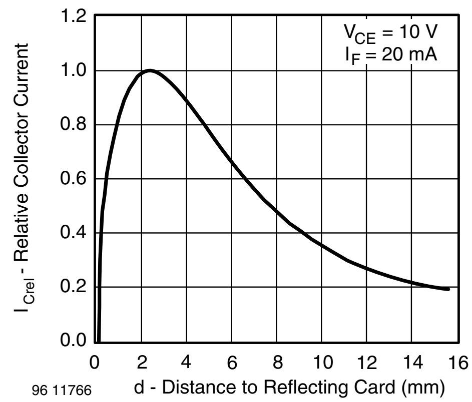
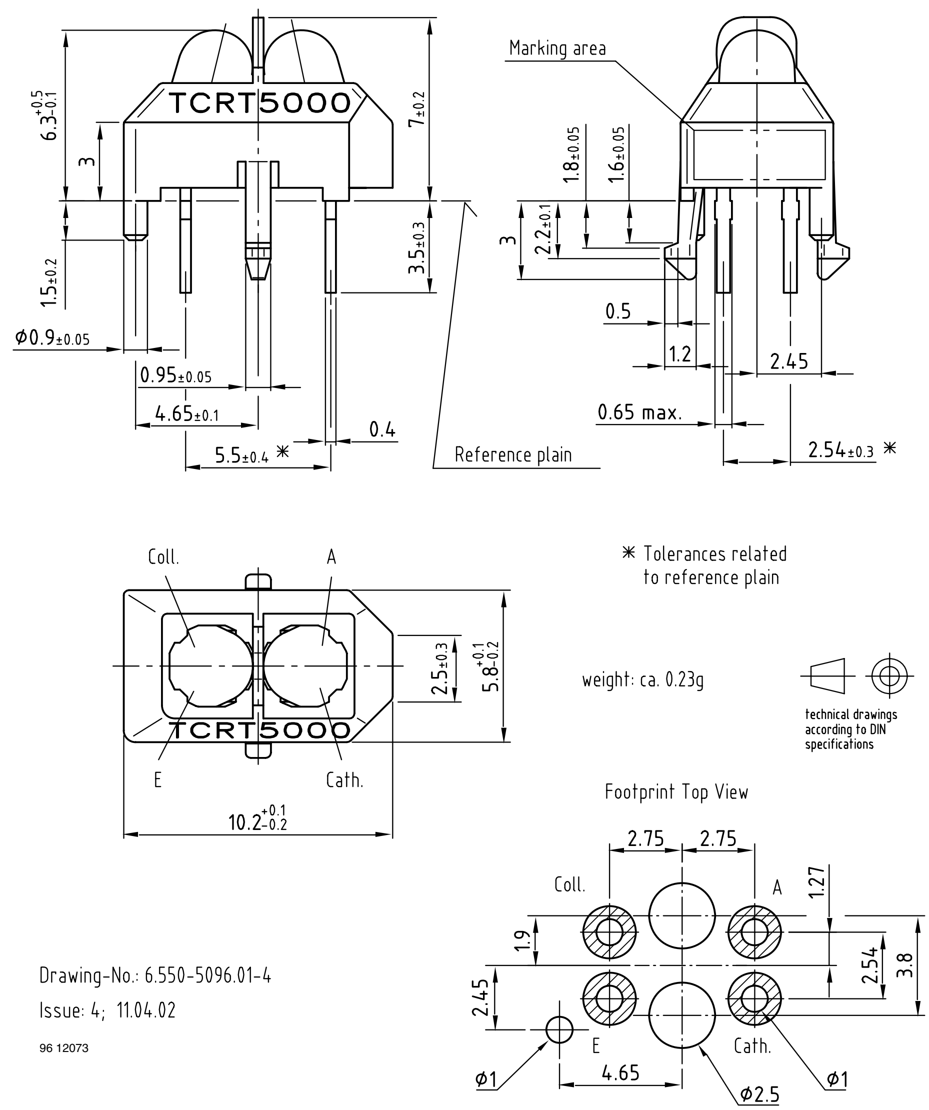
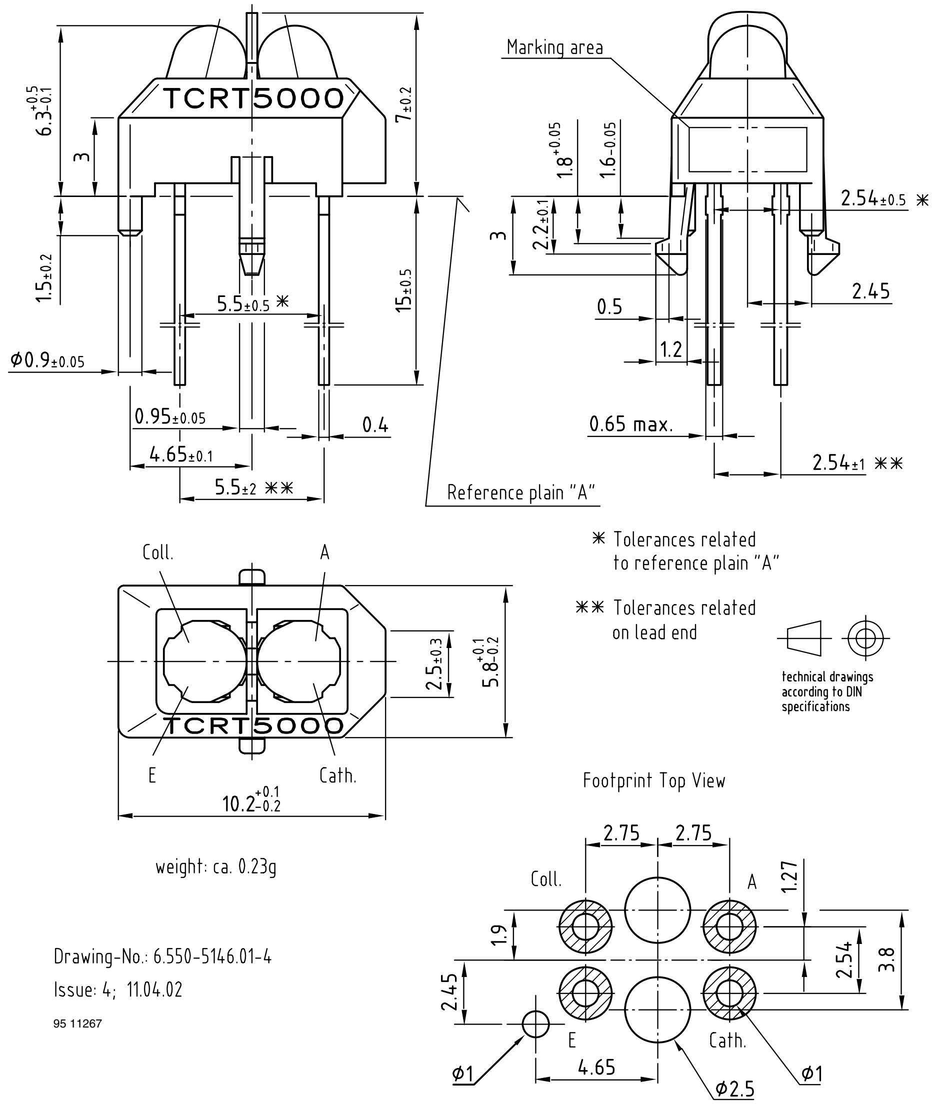


Fig. 9 - Relative Collector Current vs. Distance

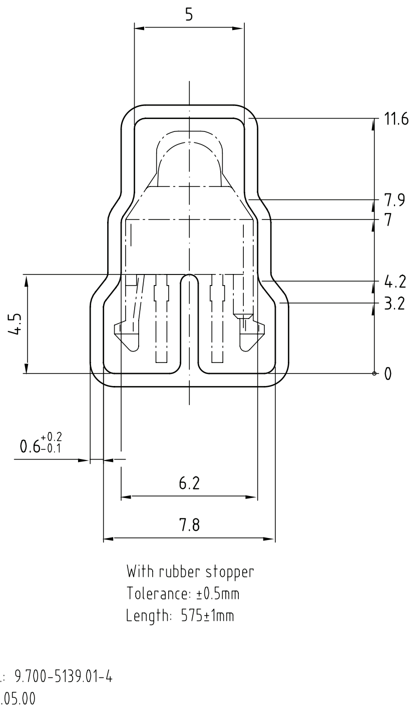
PACKAGE DIMENSIONS in millimeters, **TCRT5000**



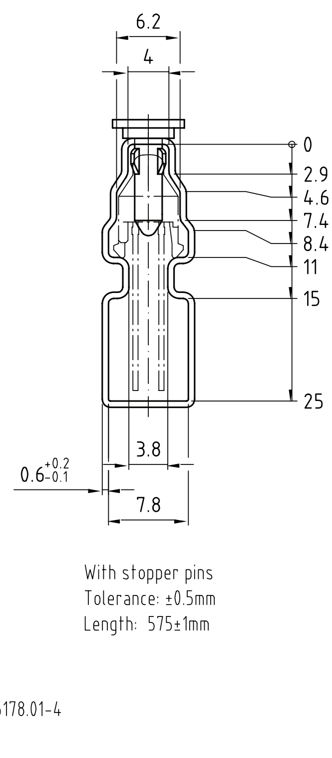
PACKAGE DIMENSIONS in millimeters, **TCRT5000L**



TUBE DIMENSIONS in millimeters, **TCRT5000**



TUBE DIMENSIONS in millimeters, **TCRT5000L**



Packaging and Ordering Information

PART NUMBER	MOQ [1]	PCS PER TUBE	TUBE SPEC. (FIGURE)	CONSTITUENTS (FORMS)
CNY70	4000	80	1	28
TCPT1300X01	2000	Reel	[2]	29
TCRT1000	1000	Bulk	-	26
TCRT1010	1000	Bulk	-	26
TCRT5000	4500	50	2	27
TCRT5000L	2400	48	3	27
TCST1030	5200	65	5	24
TCST1030L	2600	65	6	24
TCST1103	1020	85	4	24
TCST1202	1020	85	4	24
TCST1230	4800	60	7	24
TCST1300	1020	85	4	24
TCST2103	1020	85	4	24
TCST2202	1020	85	4	24
TCST2300	1020	85	4	24
TCST5250	4860	30	8	24
TCUT1300X01	2000	Reel	[2]	29
TCZT8020-PAER	2500	Bulk	-	22

Notes
[1] MOQ: minimum order quantity
[2] Please refer to datasheets

TUBE SPECIFICATION FIGURES

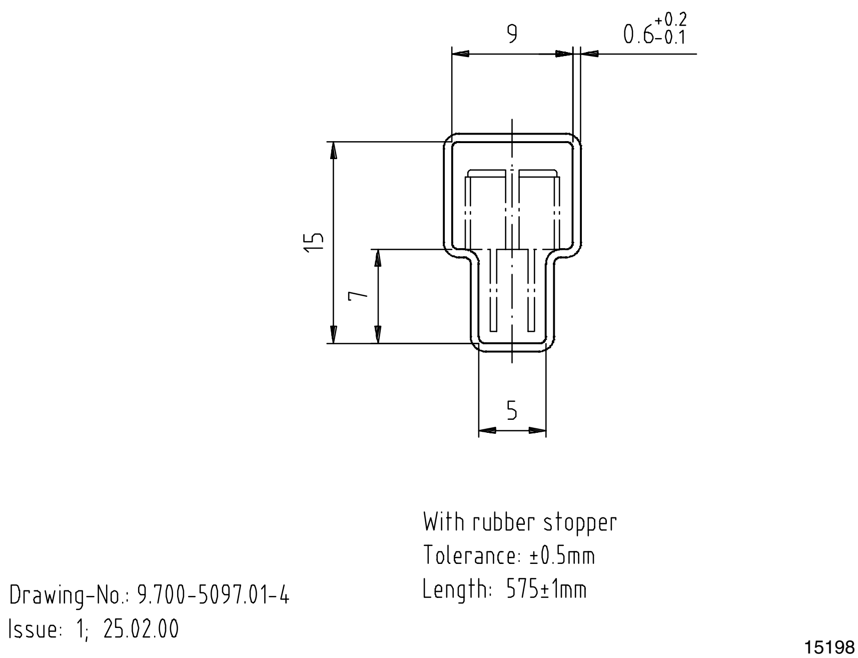
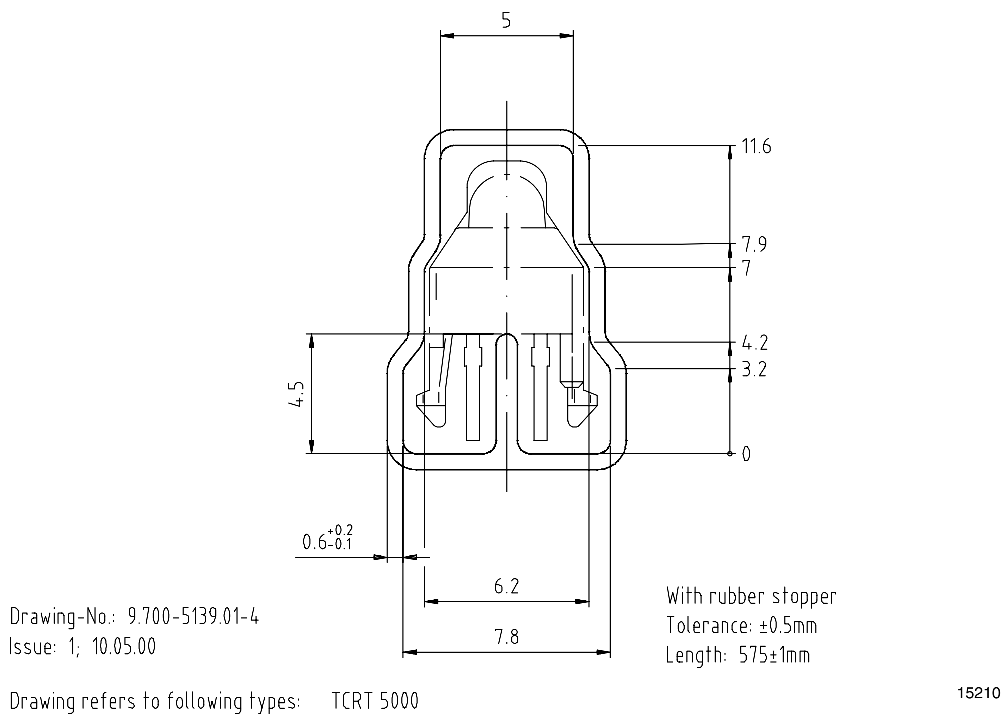
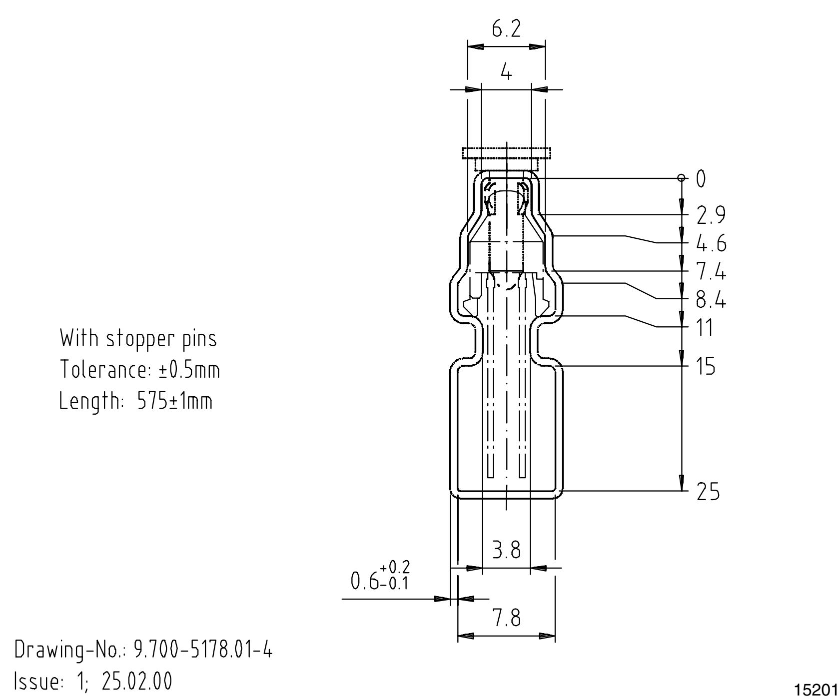


Fig. 1



5
11.6
7.9
7
4.2
3.2
0
4.5
0.6 +0.2 -0.1
6.2
7.8
Drawing-No.: 9.700-5139.01-4
Issue: 1; 10.05.00
With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm
Drawing refers to following types: TCRT 5000
15210

Fig. 2



6.2
4
0
2.9
4.6
7.4
8.4
11
15
25
With stopper pins
Tolerance: ±0.5mm
Length: 575±1mm
3.8
0.6 +0.2 -0.1
7.8
Drawing-No.: 9.700-5178.01-4
Issue: 1; 25.02.00
15201

Fig. 3

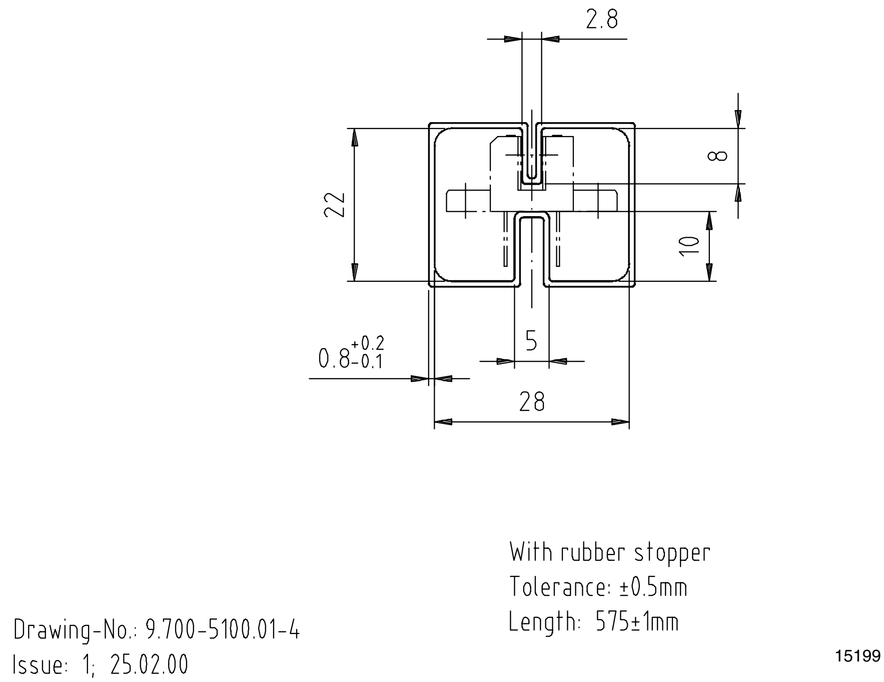

2.8
8
22
10
5
0.8 +0.2 -0.1
28
With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm
Drawing-No.: 9.700-5100.01-4
Issue: 1; 25.02.00
15199

Fig. 4

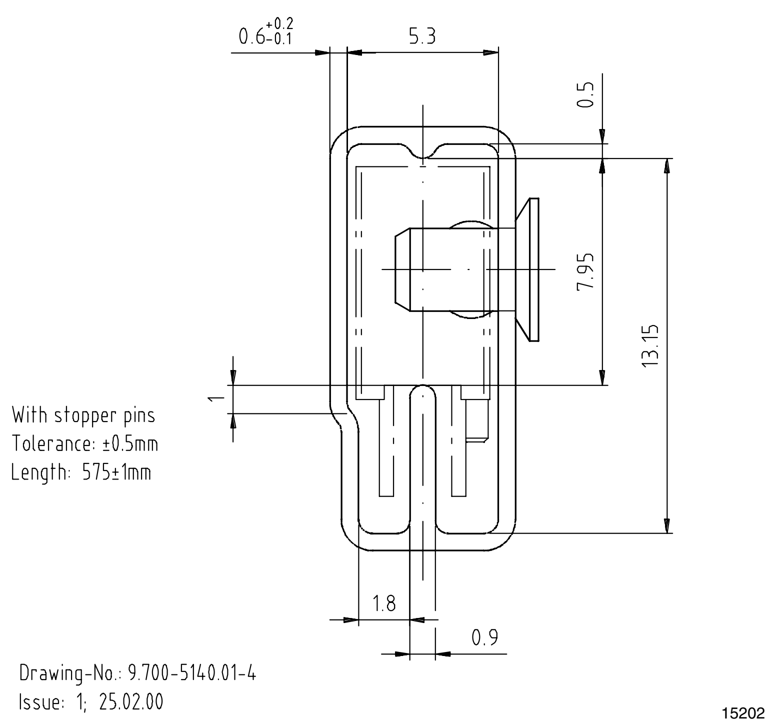
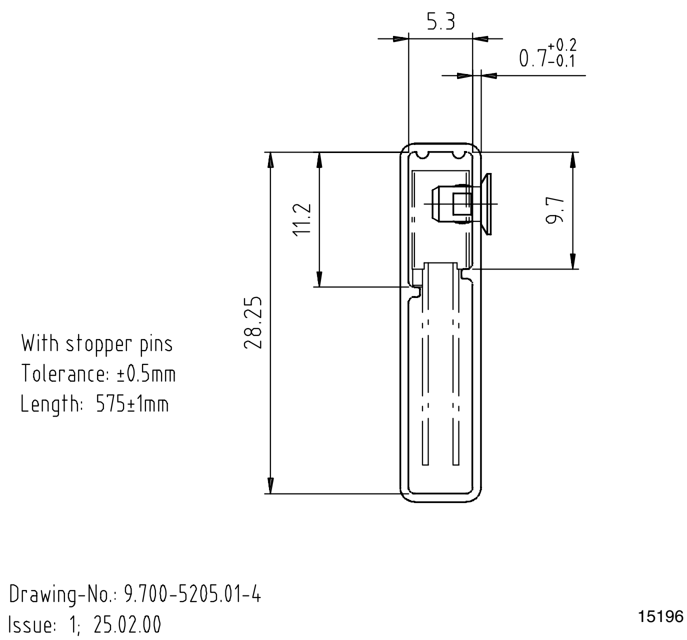
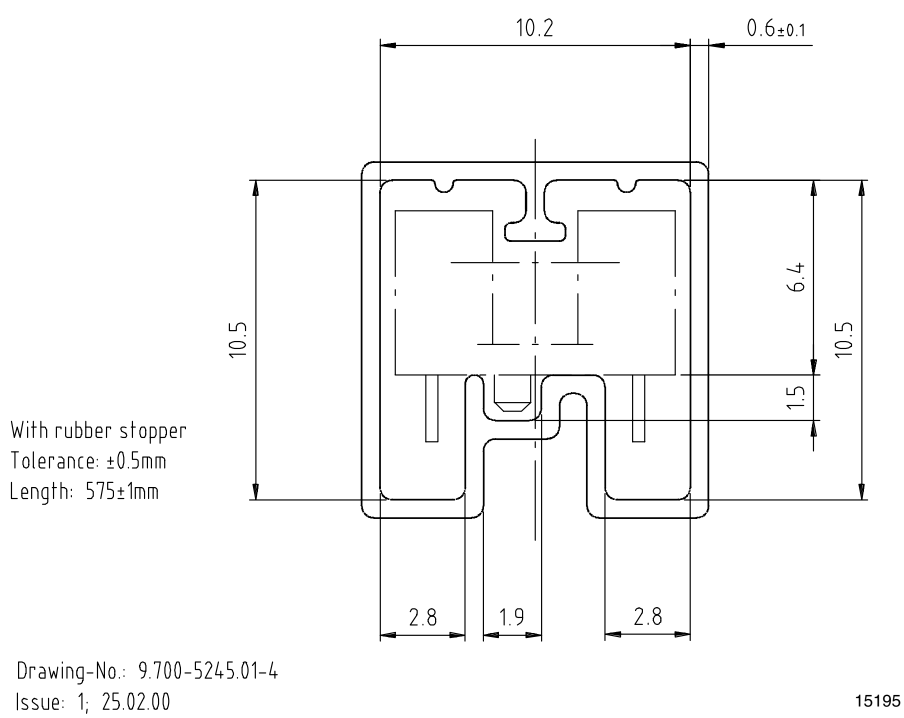

0.6 +0.2 -0.1
5.3
0.5
7.95
13.15
1
With stopper pins
Tolerance: ±0.5mm
Length: 575±1mm
1.8
0.9
Drawing-No.: 9.700-5140.01-4
Issue: 1; 25.02.00
15202

Fig. 5



5.3
0.7+0.2 -0.1
11.2
9.7
28.25
With stopper pins
Tolerance: ±0.5mm
Length: 575±1mm
Drawing-No.: 9.700-5205.01-4
Issue: 1; 25.02.00
15196

Fig. 6



10.2
0.6±0.1
10.5
6.4
10.5
1.5
With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm
2.8
1.9
2.8
Drawing-No.: 9.700-5245.01-4
Issue: 1; 25.02.00
15195

Fig. 7

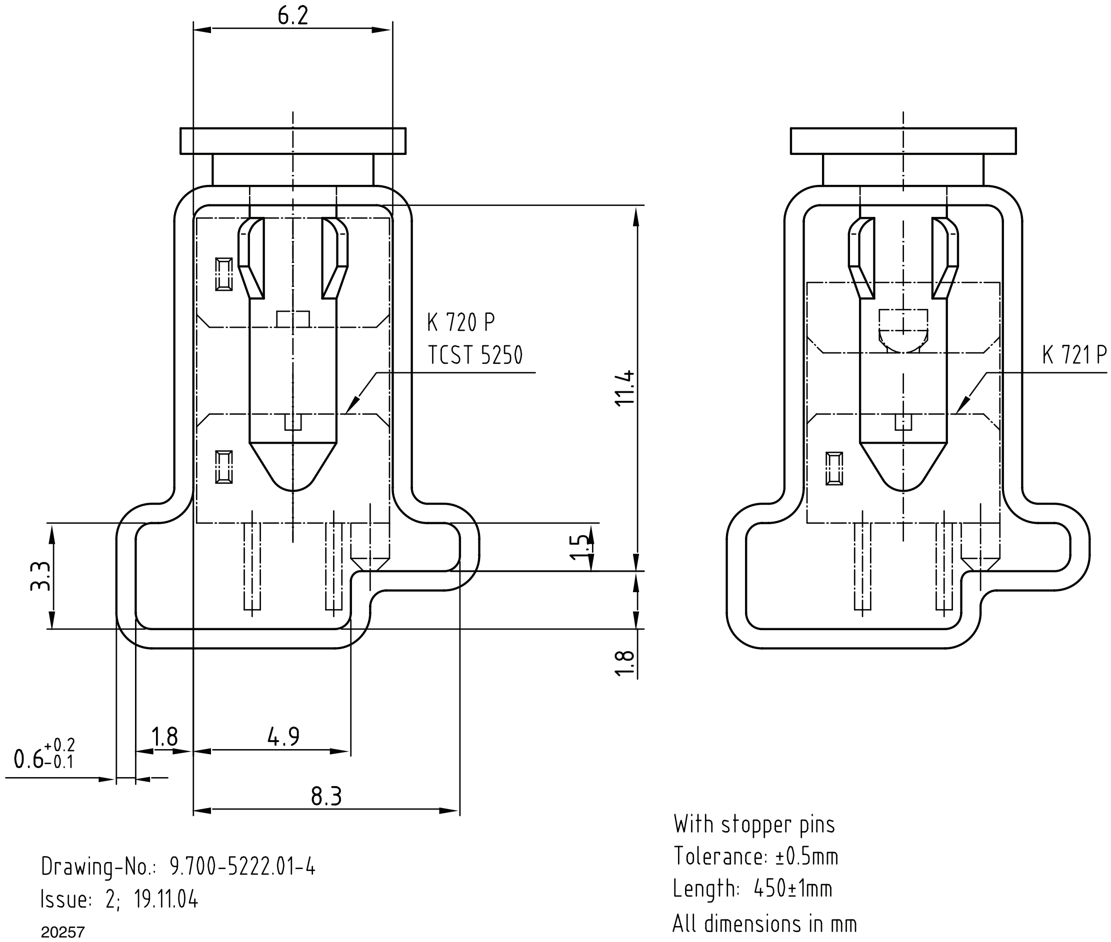

6.2
K 720 P
TCST 5250
K 721 P
11.4
1.5
3.3
1.8
$0.6^{+0.2}_{-0.1}$
1.8
4.9
8.3
Drawing-No.: 9.700-5222.01-4
Issue: 2; 19.11.04
20257
With stopper pins
Tolerance: ±0.5mm
Length: 450±1mm
All dimensions in mm

Fig. 8

Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay's knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer's responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and / or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer's technical experts. Product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein.

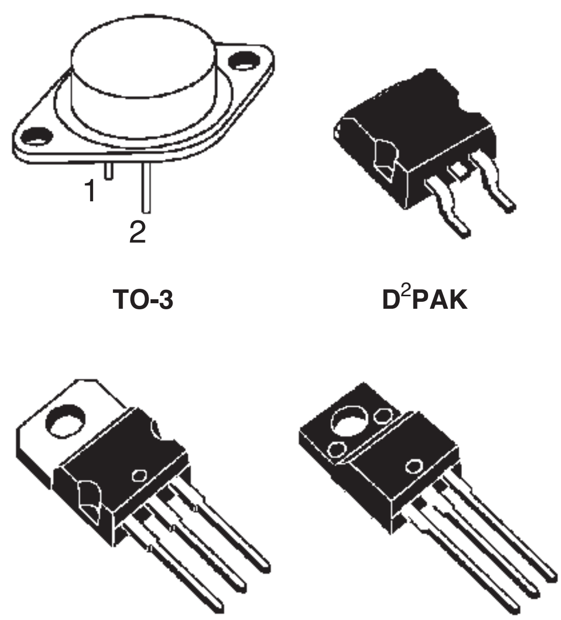
Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

L7800
SERIES

POSITIVE VOLTAGE REGULATORS

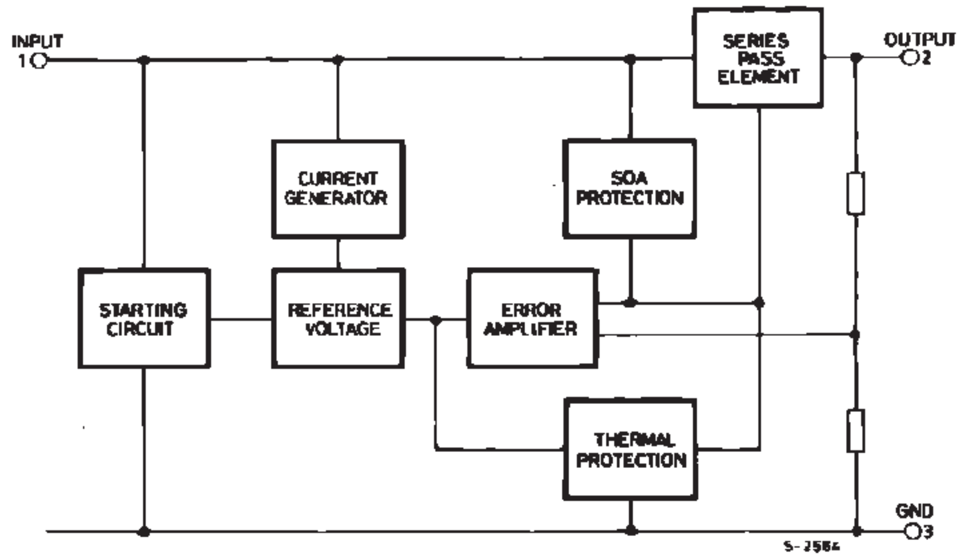
- OUTPUT CURRENT UP TO 1.5 A
- OUTPUT VOLTAGES OF 5; 5.2; 6; 8; 8.5; 9; 12; 15; 18; 24V
- THERMAL OVERLOAD PROTECTION
- SHORT CIRCUIT PROTECTION
- OUTPUT TRANSITION SOA PROTECTION



DESCRIPTION

The L7800 series of three-terminal positive regulators is available in TO-220 TO-220FP TO-3 and D^2PAK packages and several fixed output voltages, making it useful in a wide range of applications.These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

BLOCK DIAGRAM



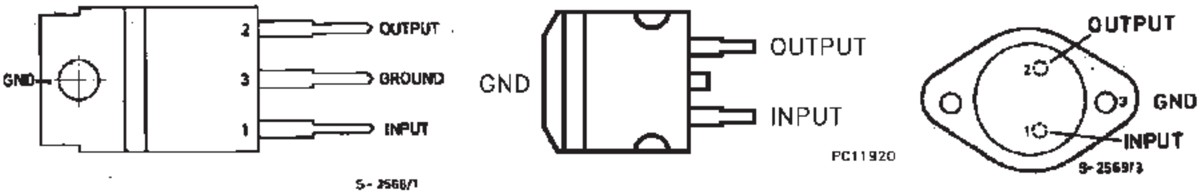
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_i	DC Input Voltage (for V_O = 5 to 18V) (for V_O = 20, 24V)	35 40	V V
I_o	Output Current	Internally limited	
P_{tot}	Power Dissipation	Internally limited	
T_{op}	Operating Junction Temperature Range (for **L7800**) (for **L7800C**)	-55 to 150 0 to 150	^oC ^oC
T_{stg}	Storage Temperature Range	-65 to 150	^oC

THERMAL DATA

Symbol	Parameter	D^2PAK	TO-220	TO-220FP	TO-3	Unit
$R_{thj-case}$ $R_{thj-amb}$	Thermal Resistance Junction-case Max Thermal Resistance Junction-ambient Max	3 62.5	3 50	5 60	4 35	$^oC/W$ $^oC/W$

CONNECTION DIAGRAM AND ORDERING NUMBERS (top view)

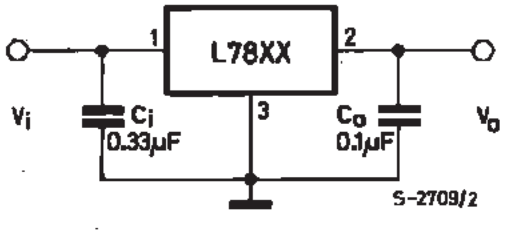


TO-220 & TO-220FP **D^2PAK** **TO-3**

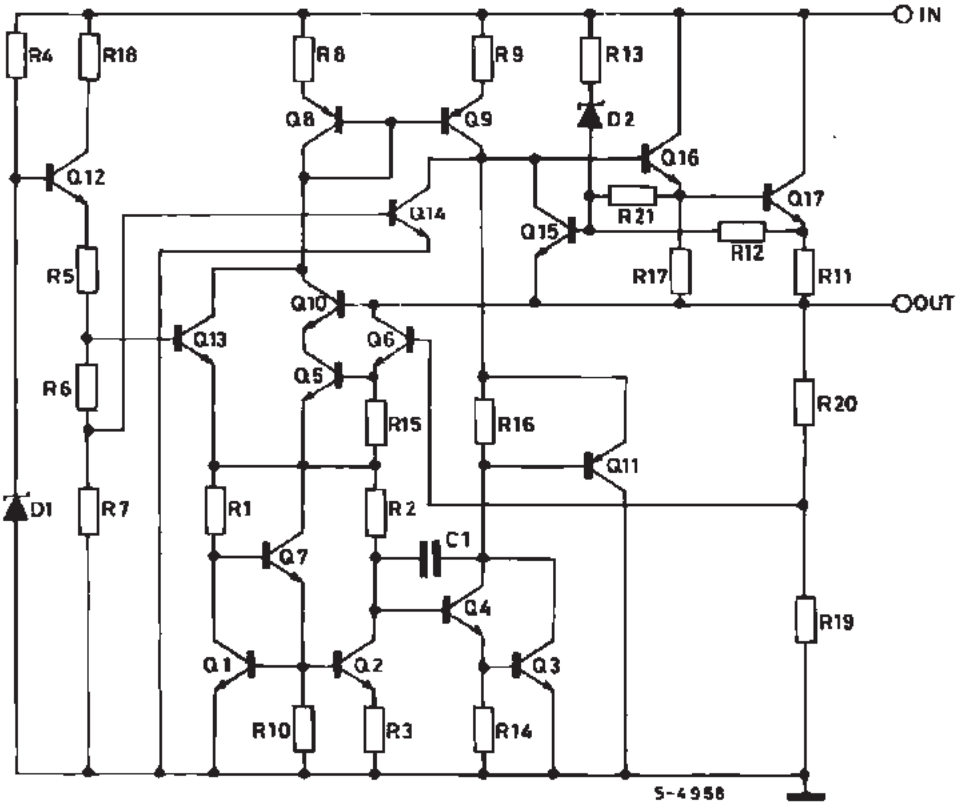
Type	TO-220	D^2PAK (*)	TO-220FP	TO-3	Output Voltage
L7805				L7805T	5V
L7805C	L7805CV	L7805CD2T	L7805CP	L7805CT	5V
L7852C	L7852CV	L7852CD2T	L7852CP	L7852CT	5.2V
L7806				L7806T	6V
L7806C	L7806CV	L7806CD2T	L7806CP	L7806CT	6V
L7808				L7808T	8V
L7808C	L7808CV	L7808CD2T	L7808CP	L7808CT	8V
L7885C	L7885CV	L7885CD2T	L7885CP	L7885CT	8.5V
L7809C	L7809CV	L7809CD2T	L7809CP	L7809CT	9V
L7812				L7812T	12V
L7812C	L7812CV	L7812CD2T	L7812CP	L7812CT	12V
L7815				L7815T	15V
L7815C	L7815CV	L7815CD2T	L7815CP	L7815CT	15V
L7818				L7818T	18V
L7818C	L7818CV	L7818CD2T	L7818CP	L7818CT	18V
L7820				L7820T	20V
L7820C	L7820CV	L7820CD2T	L7820CP	L7820CT	20V
L7824				L7824T	24V
L7824C	L7824CV	L7824CD2T	L7824CP	L7824CT	24V

(*) AVAILABLE IN TAPE AND REEL WITH "-TR" SUFFIX

APPLICATION CIRCUIT



SCHEMATIC DIAGRAM



TEST CIRCUITS

Figure 1 : DC Parameter

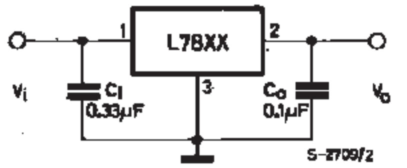


Figure 2 : Load Regulation.

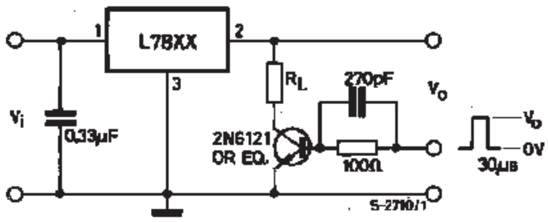
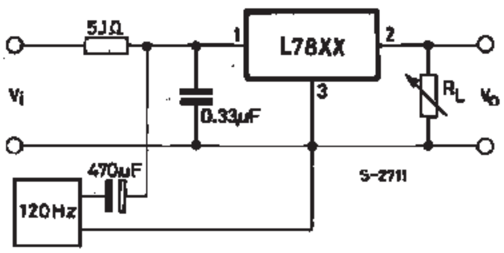


Figure 3 : Ripple Rejection.



ELECTRICAL CHARACTERISTICS FOR L7805 (refer to the test circuits, T_j = -55 to 150 ^oC, V_i = 10V, I_o = 500 mA, C_i = 0.33 μF, C_o = 0.1 μF unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_o	Output Voltage	T_j = 25 ^oC	4.8	5	5.2	V
V_o	Output Voltage	I_o = 5 mA to 1 A $P_o \leq$ 15 W V_i = 8 to 20 V	4.65	5	5.35	V
ΔV_o*	Line Regulation	V_i = 7 to 25 V T_j = 25 ^oC V_i = 8 to 12 V T_j = 25 ^oC		3 1	50 25	mV mV
ΔV_o*	Load Regulation	I_o = 5 to 1500 mA T_j = 25 ^oC I_o = 250 to 750 mA T_j = 25 ^oC			100 25	mV mV
I_d	Quiescent Current	T_j = 25 ^oC			6	mA
ΔI_d	Quiescent Current Change	I_o = 5 to 1000 mA			0.5	mA
ΔI_d	Quiescent Current Change	V_i = 8 to 25 V			0.8	mA
$\frac{\Delta V_o}{\Delta T}$	Output Voltage Drift	I_o = 5 mA		0.6		mV/^oC
eN	Output Noise Voltage	B = 10Hz to 100KHz T_j = 25 ^oC			40	μV/V_O
SVR	Supply Voltage Rejection	V_i = 8 to 18 V f = 120 Hz	68			dB
V_d	Dropout Voltage	I_o = 1 A T_j = 25 ^oC		2	2.5	V
R_o	Output Resistance	f = 1 KHz		17		mΩ
I_{sc}	Short Circuit Current	V_i = 35 V T_j = 25 ^oC		0.75	1.2	A
I_{scp}	Short Circuit Peak Current	T_j = 25 ^oC	1.3	2.2	3.3	A

ELECTRICAL CHARACTERISTICS FOR L7806 (refer to the test circuits, T_j = -55 to 150 ^oC, V_i = 15V, I_o = 500 mA, C_i = 0.33 μF, C_o = 0.1 μF unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_o	Output Voltage	T_j = 25 ^oC	5.75	6	6.25	V
V_o	Output Voltage	I_o = 5 mA to 1 A $P_o \leq$ 15 W V_i = 9 to 21 V	5.65	6	6.35	V
ΔV_o*	Line Regulation	V_i = 8 to 25 V T_j = 25 ^oC V_i = 9 to 13 V T_j = 25 ^oC			60 30	mV mV
ΔV_o*	Load Regulation	I_o = 5 to 1500 mA T_j = 25 ^oC I_o = 250 to 750 mA T_j = 25 ^oC			100 30	mV mV
I_d	Quiescent Current	T_j = 25 ^oC			6	mA
ΔI_d	Quiescent Current Change	I_o = 5 to 1000 mA			0.5	mA
ΔI_d	Quiescent Current Change	V_i = 9 to 25 V			0.8	mA
$\frac{\Delta V_o}{\Delta T}$	Output Voltage Drift	I_o = 5 mA		0.7		mV/^oC
eN	Output Noise Voltage	B = 10Hz to 100KHz T_j = 25 ^oC			40	μV/V_O
SVR	Supply Voltage Rejection	V_i = 9 to 19 V f = 120 Hz	65			dB
V_d	Dropout Voltage	I_o = 1 A T_j = 25 ^oC		2	2.5	V
R_o	Output Resistance	f = 1 KHz		19		mΩ
I_{sc}	Short Circuit Current	V_i = 35 V T_j = 25 ^oC		0.75	1.2	A
I_{scp}	Short Circuit Peak Current	T_j = 25 ^oC	1.3	2.2	3.3	A

* Load and line regulation are specified at constant junction temperature. Changes in V_o due to heating effects must be taken into account separately. Pulce testing with low duty cycle is used.

FS90--Micro 9g servo for Air plane

Model No.: FS90

Product Description

Features

-Micro analog plastic gears servo

-Operating Voltage:4.8-6Volts

-Interface: (like JR)

-Wire length: 20cm

Power	4.8V	6V
Speed	0.12sec/60degree	0.07sec/60degree
Torque	1.3kg.cm/18.9oz.in	
Weight	9g(0.32oz)	
Size	23.2*12.5*22.0mm	
Application	for Air plane	

Connection description

Orange = Signal input

Red = +5V

Brown = 0V

Typical Signals

MIN_WIDTH 544 // shortest pulse sent to a servo 0°

MAX_WIDTH 2400 // longest pulse sent to a servo 180°

NEUTRAL_PULSE_WIDTH 1500 // MID pulse width when servo is at 90°

REFRESH_INTERVAL 20000 // min time to refresh in microseconds

产品规格书

Specification of Product

产品名称：6V 1.5kg模拟连续转舵机 Product Name：6V 1.5kg.cm Analog Continuous rotation Servo
产品型号 Model No.FS90R

1.使用环境条件 Apply Environmental Condition

No.	Item	Specification
1-1	存储温度 Storage Temperature Range	-30℃～80℃
1-2	运行温度 Operating Temperature Range	-15℃～70℃

2.测试环境 Standard Test Environment

No.	Item	Specification
2-1	温度 Temperature range	25℃ ±5℃
2-2	湿度 Humidity range	65%±10%

3.机械特性 Mechanical Specification

No.	Item	Specification
3-1	尺寸 Size	A：23.2mm B：12.5mm C：22mm
3-2	重量 Weight	9g ±0.2
3-3	齿轮类型 Gear type	Plastic Gear(Nylon & POM)
3-6	机构极限角度 Limit angle	NO limit
3-7	轴承 Bearing	NO Ball bearings
3-8	出力轴 Horn gear spline	20T(4.8mm)
3-9	摆臂 Horn type	Plastic,POM
3-10	外壳 Case	Nylon & Fiberglass
3-11	舵机线 Connector wire	200mm ±5 mm
3-12	马达 Motor	Metal brush motor
3-13	防水性能 Splash water resistance	NO

4.电气特性 Electrical Specification (Function of the Performance)

No.	工作电压 Operating Voltage Range	4.8V	6V
4-1*	静态电流 Idle current(at stopped)	5mA	6mA
4-2*	空载速度 No load speed	110RPM	130RPM
4-3*	空载电流 Runnig current(at no load)	100 mA	120 mA
4-4	堵转扭矩 Peak stall torque	1.3kg.cm	1.5kg.cm
		18.09oz.in	20.86oz.in
4-5	堵转电流 Stall current	550mA	650mA

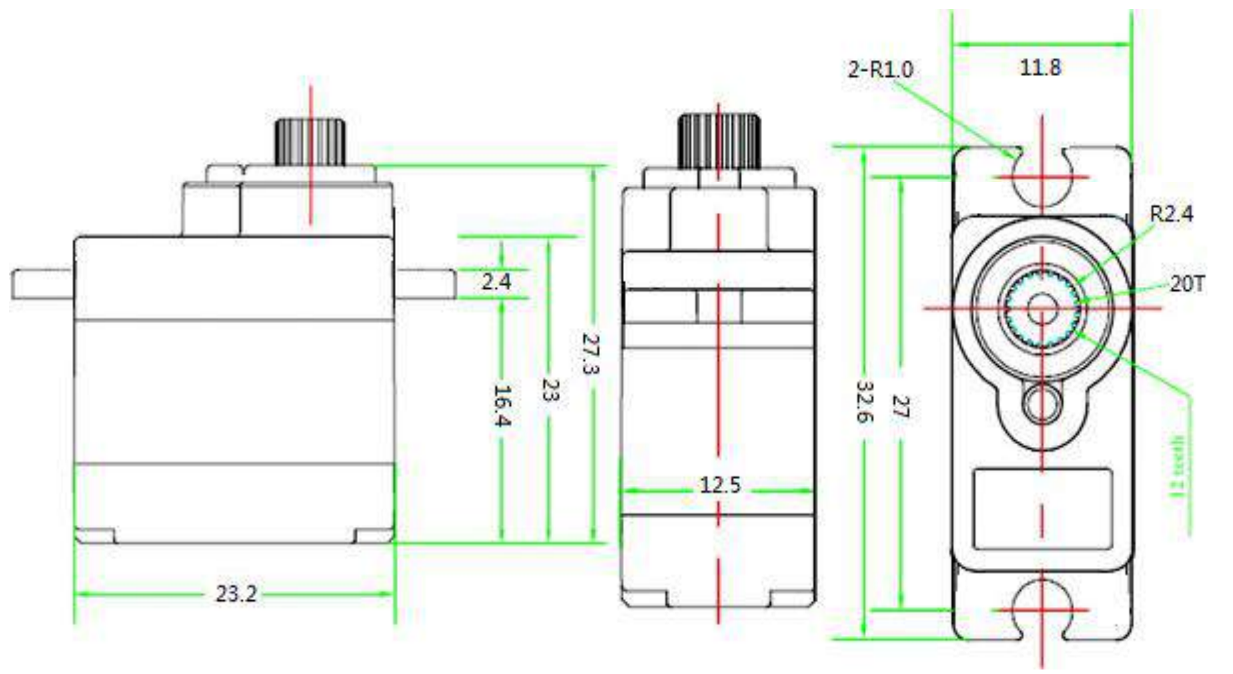
Note: "*"definition is average value when the servo runing with no load

产品名称：6V 1.5kg模拟连续转舵机 Product Name：6V 1.5kg.cm Analog Continuous rotation Servo
产品型号 Model No.FS90R

5.控制特性 Control Specification:

No.	Item	Specification
5-1	控制信号 Command signal	Pulse width modification
5-2	放大器类型 Amplifier type	Analog comparator
5-3	脉冲宽度范围 Pulse width range	700～2300μsec
5-4	停止位置 Stop position	1500（±45） μsec
5-5	旋转角度 Running degree	NO limit Continuous rotation
5-6	死区宽度 Dead band width	90 μsec
5-7	旋转方向 Rotating direction	CW(when 1500～700 μsec) CCW(when 1500～2300 μsec)

6.外形图 The Drawings



IR Emitter and Detector

Product Data Sheet

LTE-5208A

Spec No.: DS-50-92-0034

Effective Date: 09/26/2009

Revision: B

LITE-ON DCC

RELEASE

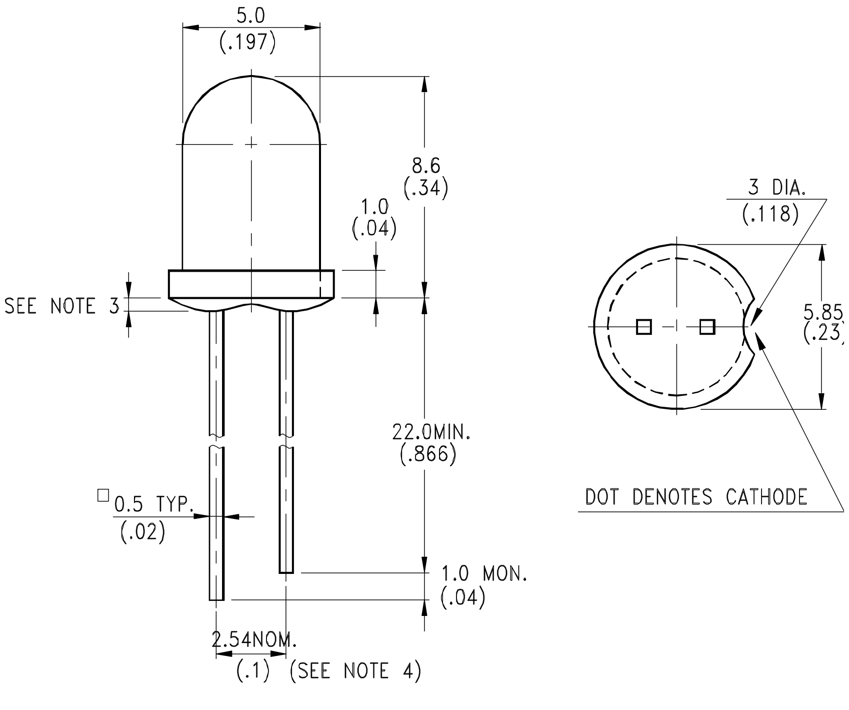
BNS-OD-FC001/A4

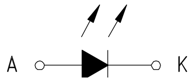
LITE-ON Technology Corp. / Optoelectronics
No.90,Chien 1 Road, Chung Ho, New Taipei City 23585, Taiwan, R.O.C.
Tel: 886-2-2222-6181 Fax: 886-2-2221-1948 / 886-2-2221-0660
http://www.liteon.com/opto

FEATURES

* SELECTED TO SPECIFIC ON-LINE INTENSITY AND RADIANT INTENSITY RANGES
* LOW COST MINIATURE PLASTIC END LOOKING PACKAGE
* MECHANICALLY AND SPECTRALLY MATCHED TO THE LTR-3208 SERIES OF PHOTOTRANSISTOR
* CLEAR TRANSPARENT COLOR PACKAGE

PACKAGE DIMENSIONS





NOTES:
1. All dimensions are in millimeters (inches).
2. Tolerance is ±0.25mm(.010") unless otherwise noted.
3. Protruded resin under flange is 1.5mm(.059") max.
4. Lead spacing is measured where the leads emerge from the package.
5. Specifications are subject to change without notice.

ABSOLUTE MAXIMUM RATINGS AT TA=25℃

PARAMETER	MAXIMUM RATING	UNIT
Power Dissipation	150	mW
Peak Forward Current (300pps, 10 μs pulse)	2	A
Continuous Forward Current	100	mA
Reverse Voltage	5	V
Operating Temperature Range	-40℃ to + 85℃	
Storage Temperature Range	-55℃ to + 100℃	
Lead Soldering Temperature [1.6mm(.063") From Body]	260℃ for 5 Seconds	

ELECTRICAL OPTICAL CHARACTERISTICS AT TA=25℃

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITION	BIN NO.
Aperture Radiant Incidence	Ee	0.44		0.96	mW/cm²	I_F = 20mA	BIN A
		0.64		1.20			BIN B
		0.80		1.68			BIN C
		1.12					BIN D
Radiant Intensity	I_E	3.31		7.22	mW/sr	I_F = 20mA	BIN A
		4.81		9.02			BIN B
		6.02		12.63			BIN C
		8.42					BIN D
Peak Emission Wavelength	λ_{Peak}		940		nm	I_F = 20mA	
Spectral Line Half-Width	$\Delta\lambda$		50		nm	I_F = 20mA	
Forward Voltage	V_F		1.2	1.6	V	I_F = 20mA	
Reverse Current	I_R			100	μA	V_R = 5V	
Viewing Angle (See FIG.6)	$2\theta_{1/2}$		40		deg.		

TYPICAL ELECTRICAL / OPTICAL CHARACTERISTICS CURVES

(25℃ Ambient Temperature Unless Otherwise Noted)

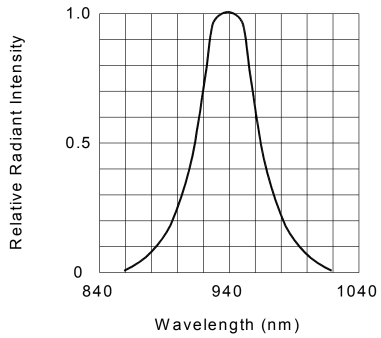


FIG.1 SPECTRAL DISTRIBUTION

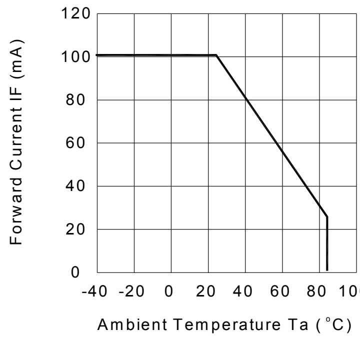


FIG.2 FORWARD CURRENT VS. AMBIENT TEMPERATURE

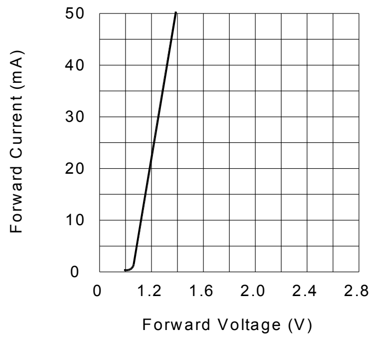


FIG.3 FORWARD CURRENT VS. FORWARD VOLTAGE

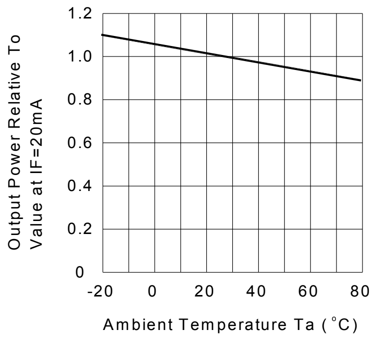


FIG.4 RELATIVE RADIANT INTENSITY VS. AMBIENT TEMPERATURE

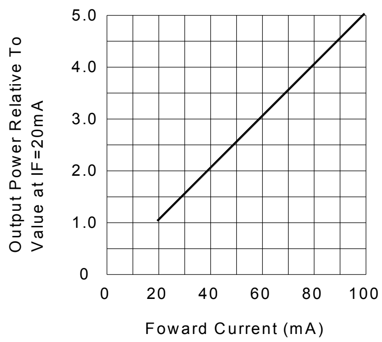


FIG.5 RELATIVE RADIANT INTENSITY VS. FORWARD CURRENT

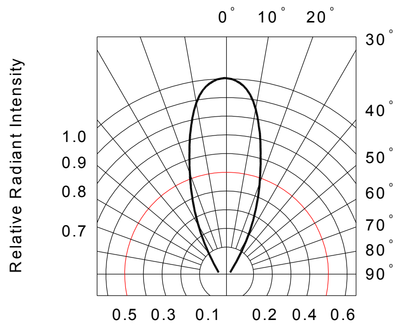


FIG.6 RADIATION DIAGRAM

IR Emitter and Detector

Product Data Sheet

LTR-3208E

Spec No.: DS-50-92-0068

Effective Date: 05/03/2000

Revision: A

LITE-ON DCC

RELEASE

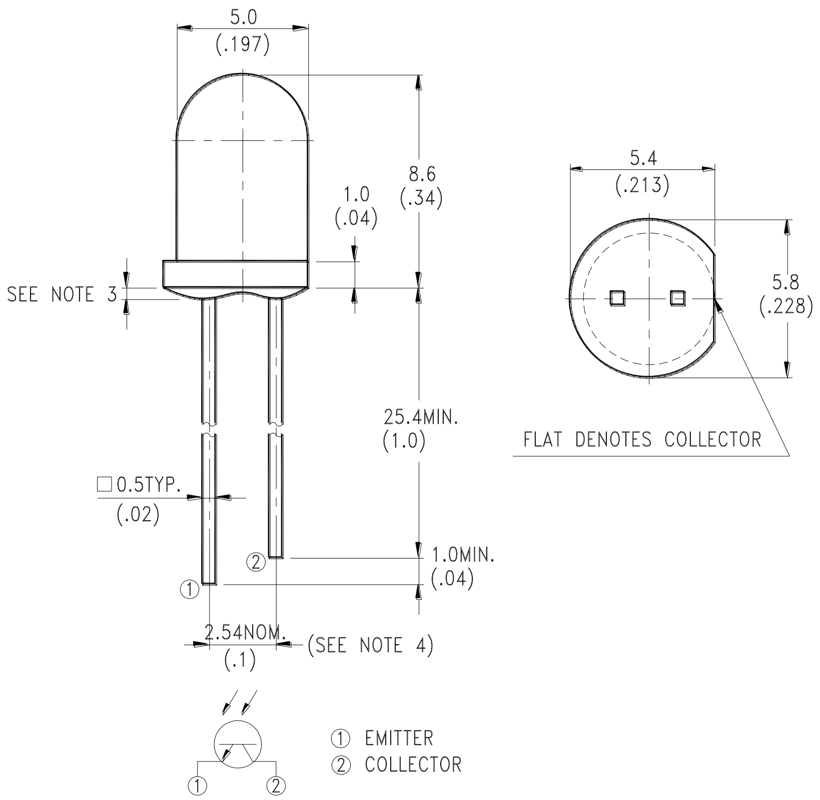
BNS-OD-FC001/A4

LITE-ON Technology Corp. / Optoelectronics
No.90,Chien 1 Road, Chung Ho, New Taipei City 23585, Taiwan, R.O.C.
Tel: 886-2-2222-6181 Fax: 886-2-2221-1948 / 886-2-2221-0660
http://www.liteon.com/opto

FEATURES

* WIDE RANGE OF COLLECTOR CURRENT
* THE LENS IS FOR HIGH SENSITIVITY
* LOW COST PLASTIC PACKAGE
* THE LTR-3208E IS A SPECIAL DARK PLASTIC PACKAGE THAT CUT THE VISIBLE LIGHT AND SUITABLE FOR THE DETECTORS OF INFRARED APPLICATIONS

PACKAGE DIMENSIONS



NOTES:
1. All dimensions are in millimeters (inches).
2. Tolerance is ±0.25mm(.010") unless otherwise noted.
3. Protruded resin under flange is 1.5mm(.059") max.
4. Lead spacing is measured where the leads emerge from the package.
5. Specifications are subject to change without notice.

ABSOLUTE MAXIMUM RATINGS AT TA=25℃

PARAMETER	MAXIMUM RATING	UNIT
Power Dissipation	100	mW
Collector-Emitter Voltage	30	V
Emitter-Collector Voltage	5	V
Operating Temperature Range	-40℃ to + 85℃	
Storage Temperature Range	-55℃ to + 100℃	
Lead Soldering Temperature [1.6mm(.063") From Body]	260℃ for 5 Seconds	

ELECTRICAL / OPTICAL CHARACTERISTICS AT TA=25℃

PARAMETER	SYMBOL	MIN.	TYP.	MAX	UNIT	TEST CONDITION	BIN NO.
Collector-Emitter Breakdown Voltage	$V_{(BR)CEO}$	30			V	I_C = 1mA Ee = 0mW/cm²	
Emitter-Collector Breakdown Voltage	$V_{(BR)ECO}$	5			V	I_E = 100 μA Ee = 0mW/cm²	
Collector Emitter Saturation Voltage	$V_{CE(SAT)}$		0.1	0.4	V	I_C = 100 μA Ee = 1mW/cm²	
Rise Time	Tr		10		μs	V_{CC} = 5V I_C = 1mA R_L = 1KΩ	
Fall Time	Tf		15		μs		
Collector Dark Current	I_{CEO}			100	nA	V_{CE} = 10V Ee = 0mW/cm²	
On State Collector Current	$I_{C(ON)}$	0.64		1.68	mA	V_{CE} = 5V Ee = 1mW/cm² λ =940nm	BIN A
		1.12		2.16			BIN B
		1.44		2.64			BIN C
		1.76		3.12			BIN D
		2.08		3.60			BIN E
		2.40					BIN F

TYPICAL ELECTRICAL / OPTICAL CHARACTERISTICS CURVES

(25℃ Ambient Temperature Unless Otherwise Noted)

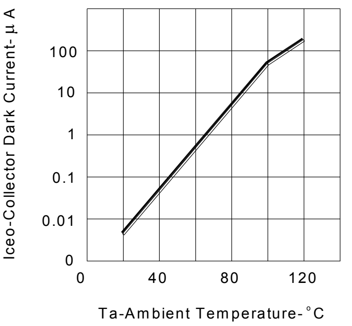


FIG.1 COLLECTOR DARK CURRENT VS AMBIENT TEMPERATURE

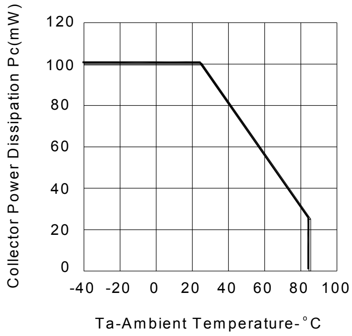


FIG.2 COLLECTOR POWER DISSIPATION VS AMBIENT TEMPERATURE

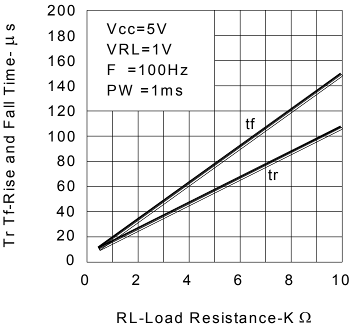


FIG.3 RISE AND FALL TIME VS LOAD RESISTANCE

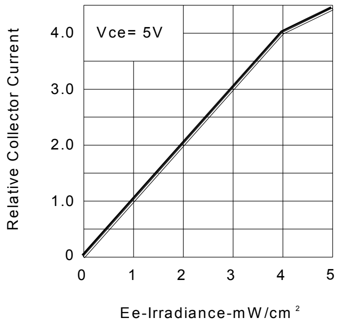


FIG.4 RELATIVE COLLECTOR CURRENT VS IRRADIANCE

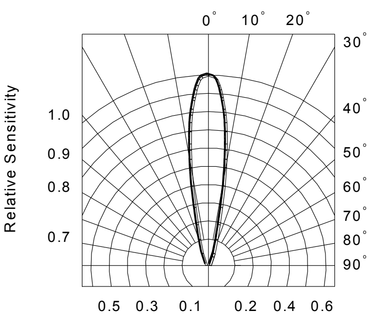


FIG.5 SENSITIVITY DIAGRAM