

UNIVERSITY OF TORONTO

AER201, JANUARY 2010

Helical Cone Deployment Machine

Author:

Ahil Ganesh (997xxxxxx)
Sebastian Kosch (997241024)
Karl Qin (997xxxxxx)

Supervisor:

Dr. Reza Emami

January 14, 2011

Chapter 1

Executive Summary

Chapter 2

Introduction

2.1 Statement of Need

We *really* need these, because traffic workers die all the time.

2.2 Goals and Objectives

2.3 Background Research

2.3.1 Existing Designs

2.3.1.1 Autocone

The Autocone 130 is a large trailer that stores up to 130 traffic cones. It uses a hydraulic system both to deploy the cones and to pick up cones standing or lying on the road. The cones are stored in a revolving framework. Within the framework, the cones are moved to the deployment arm by means of a chain conveyor system [?]. While the compactness of the revolving frame is convincing, the capacity is far higher than that required in this project. For this prototype, designing an intricate miniature hydraulic arm is thus not justifiable.

2.3.1.2 TRAF-tech ACT/M Series

TRAF-tech commercialized the cone deployment system initially developed by the California Department of Transportation. Two different implementations are available: one uses rails to deploy the cones, the other a hydraulic arm. Both rely on belt systems to move the individual cones from the stack to the deployment mechanism. Like Autocone, TRAF-tech offers systems with a comparatively high cone capacity. Conveyor belts, however, are difficult to build reliably,¹ and may not be of much help with the more flexible toy cones at hand.

¹as has been confirmed by past AER201 students.



(a) Autocone



(b) TRAF-tech ACM

Figure 2.1: Existing designs (Photo: <http://www.innovativeequip.com/>, <http://www.traftech.net/>)

2.3.2 Possible solutions for the given RFP

Chapter 3

Technical Body

3.1 Statement of Work

3.1.1 Components

3.1.1.1 Mechanical

Wheels The wheels are ...

Coils The coils move the cones.

3.1.1.2 Electronic/Sensors

Driver Board Made by Karl, this customized driver board is awesome.

Sensors Special sensors.

3.1.1.3 Microcontroller

A single microcontroller processes the sensors' inputs, computes the power delivered to the motors, and interacts with the user via a matrix keypad and a 16×2 character LCD.

Choice of model The microcontroller used is a 40-pin PIC chip by Microchip, model 18F4620. The model was chosen over model 16F877, which was given with the development board, mainly for its interrupt capabilities. The 16F series provides an external interrupt channel on pin RB0. Unfortunately, this does not allow for interrupt-based development with the development board's keypad, since the keypad's `DataReady`-flag is hardwired to RB1. Using the Interrupt-on-change feature on RB4:7 does not help; pressing the same key twice will not raise an interrupt. The same rationale explains the choice of a 40-pin model. While all functionality could have been implemented, tested and debugged on the development board with the given 16F877 series PIC, interrupt-based code is easier to debug than code littered with `goto`-based polling points.

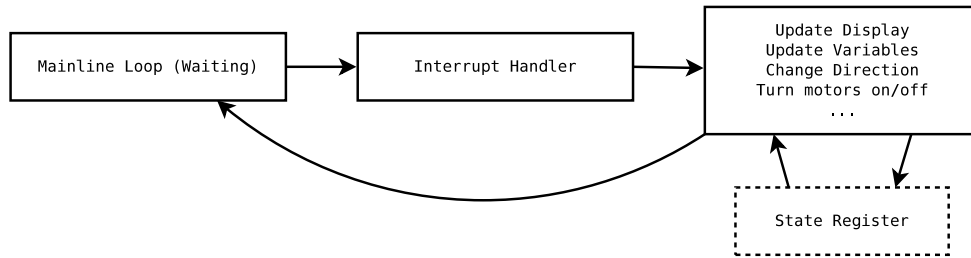


Figure 3.1: General flow of the microcontroller program

Program flow and menu On power-on, the microcontroller initializes memory, interrupt bits and I/O settings. It then waits for input on interrupt lines. [SOMETHING ABOUT POWER CONSUMPTION IN THE LOOP VS. POLLING]. The following events should raise an interrupt:

- the user presses a key on the keypad,
- the interval timer overflows to cause a cone to be deployed,
- the reflection sensor reports a hole,
- all cones have been deployed, or a distance of 3 meters was covered,
- the side reflection sensors report crossing the lane.

The interrupt service routine (ISR) will then determine the type of interrupt. A state register holds a numerical value representing the current state of the device, such as *Menu*, *Screen 3* or *Returning from deployment*. Depending on the current state, the ISR calls subroutines to handle the input. Example: during user interaction, this might mean storing the input and updating the display; during deployment, this might mean rotating the coil or decelerating the motors. It may also include changing the state register. Eventually, all subroutines will return to the **Mainline** loop to save power.

3.1.2 Functional Description

The cone deployment machine will prompt the user for input, deploy the cones, return to the user and display information about the deployment. This communication happens via a 4×4 matrix keypad (keys 0–9 and A–D) and a 16×2 character LCD with standard HD44780 interface. To make operation as easy as possible, the interaction using display and keypad should be self-explanatory to the user. Therefore, we designed a near-linear sequence of inputs and outputs that leads the user through all necessary steps. This is adequate for both novices and experienced users. Figure A.1 provides an overview.

The machine will go and dispense cones. ... bla.

3.1.3 Evaluation Plan

3.1.3.1 Process evaluation

3.1.3.2 Product evaluation

3.2 Schedule

Bar chart, bar chart, bar chart.

3.3 Budget

Chapter 4

Conclusion

Appendix A

Appendix

A.1 Figures and Tables

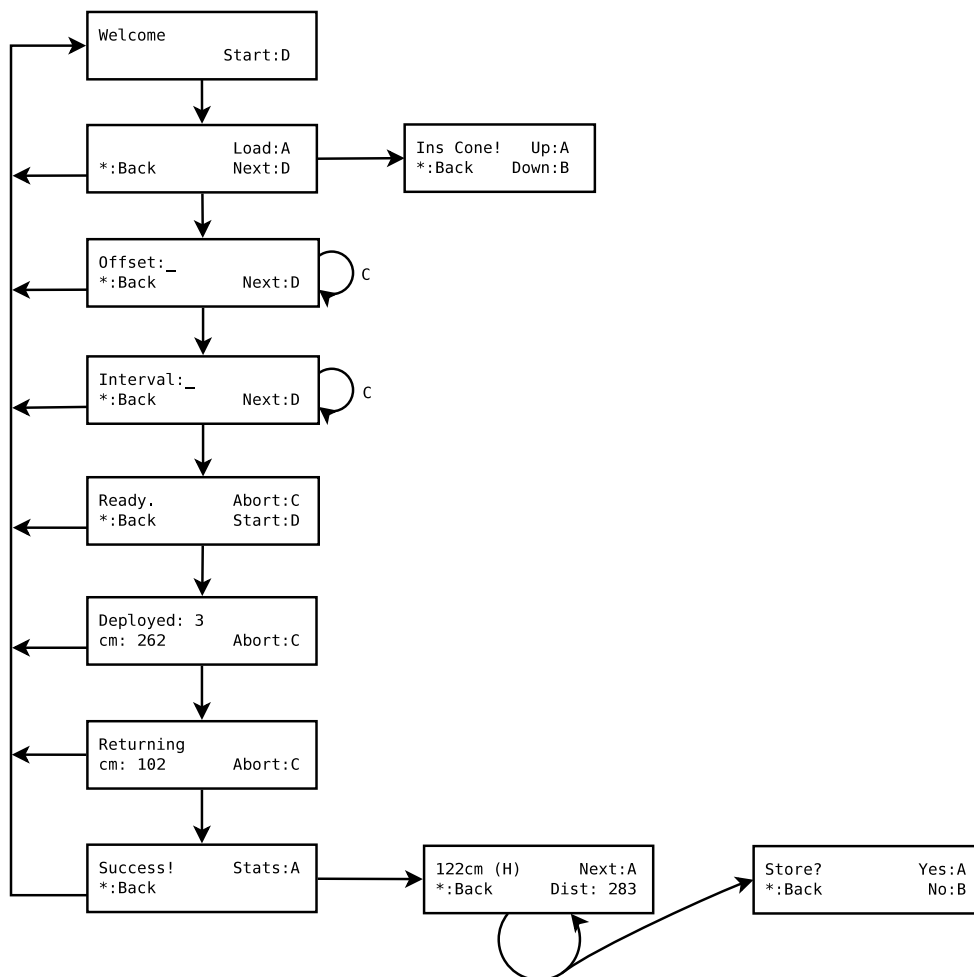


Figure A.1: Sequence of menu displays navigable by the user