Drone Design Manual

Shao Hui z5155945

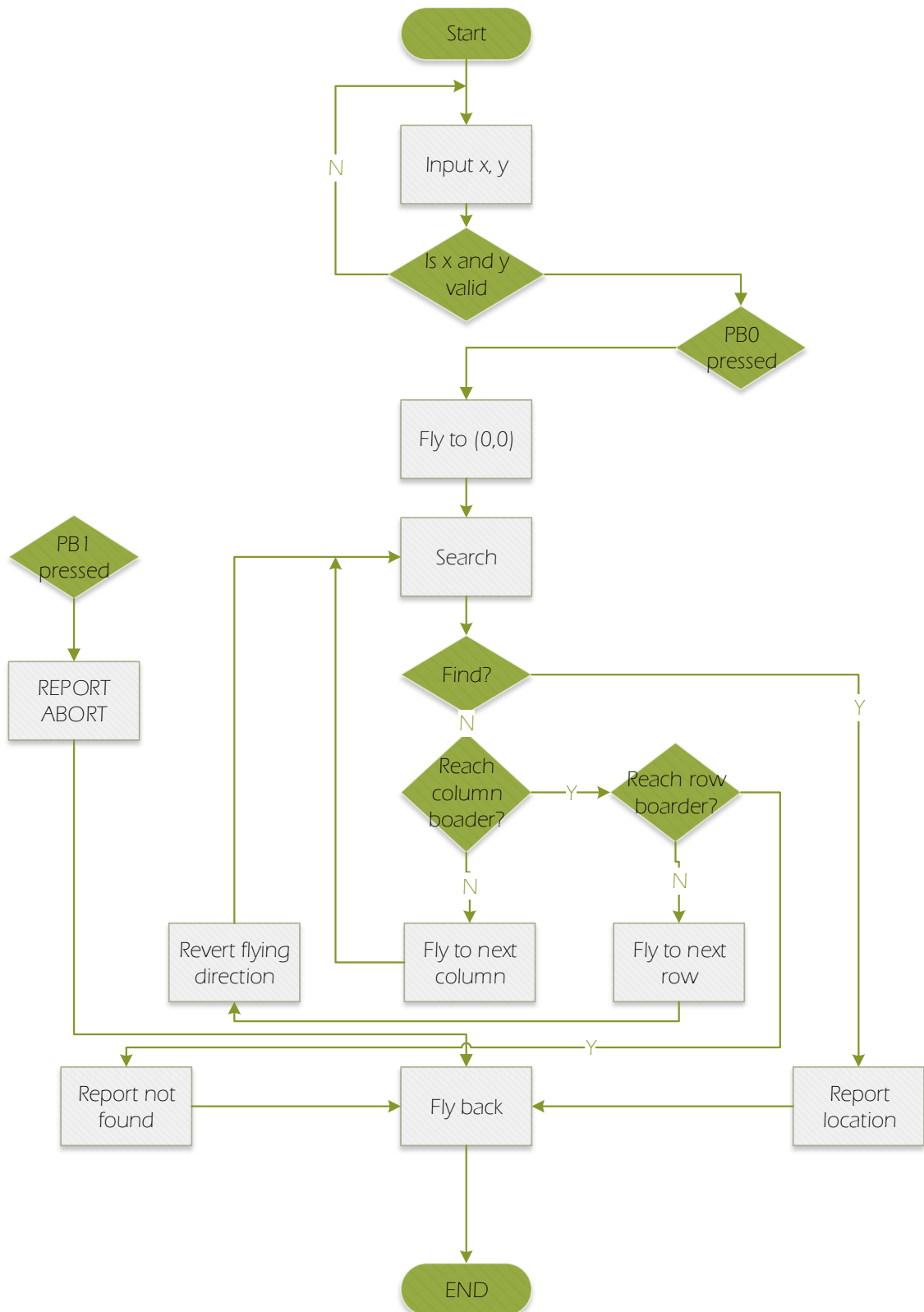# Contents

0.1 Introduction

This project simulates a drone searching for a accident scene in a mountain which is presented by a 64×64 matrix.

Drone starts searching from possition (0,0), then fly east and searching grids one by one. Once the drone reaches the boarder of the mountain, it will fly south for 1 meter, then fly to the other direction. Program keeps the drone suspend above the mountain by 1 meter to make sure every spot can be seen.

0.2 Diagram

Start

Input x, y

Is x and y valid

N

PB0 pressed

Fly to (0,0)

PB1 pressed

Search

REPORT ABORT

Find?

N

Reach column boarder?

Y

Reach row boarder?

N

N

Revert flying direction

Fly to next column

Fly to next row

Report not found

Y

Fly back

Y

Report location

END

0.3 Modules

tables.pdf

0.4 Algorithms

Algorithm 1 Fly to next position

1: try go to next position as direction shows
2: **if** border reached **then**
3:     go south
4:     **if** border reached **then**
5:         **return** -1
6:     **end if**
7:     flip direction
8: **end if**
9: **return** position height

Algorithm 2 Drone search

1: read accident position
2: **if** accident scene find **then**
3:     **return** *true*
4: **else**
5:     **return** *false*
6: **end if**

0.5   Data Structure

0.5.1   Mountain

Mountain is store in program memory as a 2 dimension array of 8bit integer.
The index of the array presents the relative position of each 1×1 grid. The value stored
in array presents the height of each position.
To get each position's address simply follow such formular.

$$addr\_position_{x,y} = addr_{Mountainstart} + x \times 63 + y$$

0.5.2   Accident scene

Accident scene is stored in data memory as a two 8bit integer, and can be modified by
user.
Sure we can store accident scene by modify mountain height(eg. plus 128), but store it
in the data memory can save it for more general use.