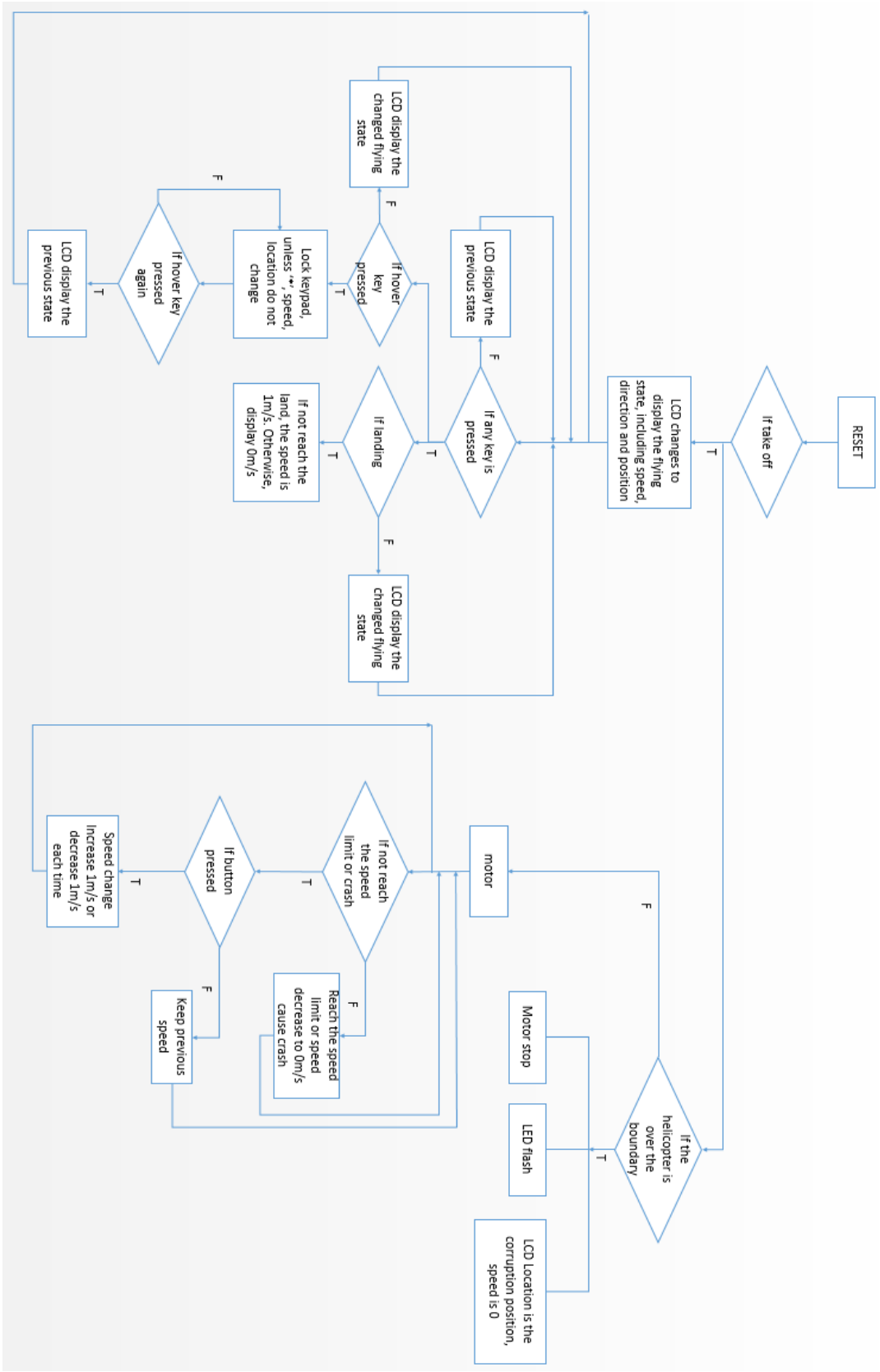
Design Manual

1. System control flow diagram



2. Data structure

Temporary register:

temp1 = r16; temp2 = r16; temp3 = r22.

These three registers are used to transfer data, and they could be used in anywhere, because they would not store a value for a long time. When after transferring data or finish a function, they could be used again.

Data memory:

(1) Location parameters: posX; posY; posZ

In data memory, these three parameters are used to store location parameters. When the location is required to change, macro transfer_to_register could fetch the value from the data memory and store the value into a register. After change the value through the temporary register, macro move_to_data could move the value from the register to the data memory.

With using timer0, every second the location would be changed according to the speed value. To record the direction change, the location is required to add or subtract the speed value every second.

(2) Direction parameter: dir

In data memory, this parameter is used to store direction value, when the keypad is pressed the direction may change, so macro transfer_to_register and macro transfer_to_data could be used to complete the process of fetching and storing direction value.

(3) Speed parameter: speed

In data memory, this parameter is used to store speed value. When PB0 or PB1 or # or * is pressed, the speed would change. In the design macro transfer_to_register, transfer_to_data and move_to_data are all used to fetch or store the value.

e.g . transfer_to_data 2,speed; set speed to 2m/s

e.g. transfer_to_register speed,temp2; transfer the speed value to register temp2

e.g. move_to_data temp2,speed; transfer the value of register temp2 to speed value

(4) State_flag

This flag is stored in the data memory. The initial value is 0. When '#' is first pressed, the helicopter starts to take off and the flag value is set to 1, which means that the helicopter is enter to the flying state. When landing ('#' is pressed again), the flag will be set to 0, and the helicopter start to land.

(5) hovering_flag

This flag is stored in the data memory. The initial value is 0. When '*' is pressed, it will check whether the flag has already set to 1. If the flag value is 0, the helicopter will start to hover. If the flag value has already set to 1, it will stop hover and continue previous action.

(6) GlobalCounter

In data memory, this parameter is used to count the seconds.(Timer0)

(7) flyingCounter

When the helicopter is safely landed, it will fetch the GlobalCounter's value and

store it into flyingCounter.

(8) hole_counter and round_counter
In order to count the speed of the motor, hole_counter is used to counter the hole and every 4 holes is countered round_counter is required to increase 1.

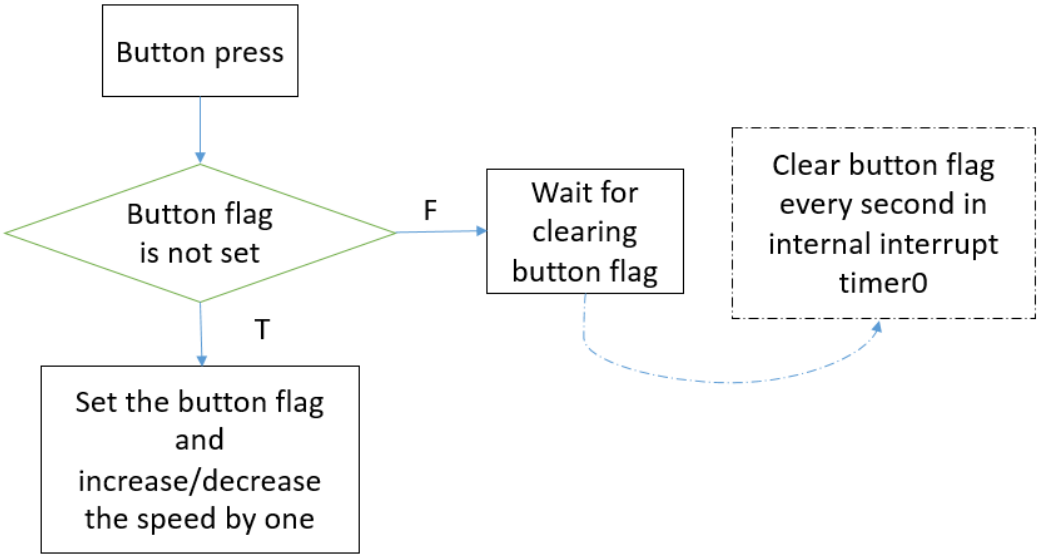
(9) button_flag
When this flag is 0, PB0 or PB1 could increase or decrease 1m/s. Once the button is pressed, the flag is set to 1. And in timer0, the flag is set to 0 every second. In this way, the button will only add or decrease 1m/s when button is pressed.

Keypad used register:

row	r18
col	r19
rmask	r20
cmask	R21

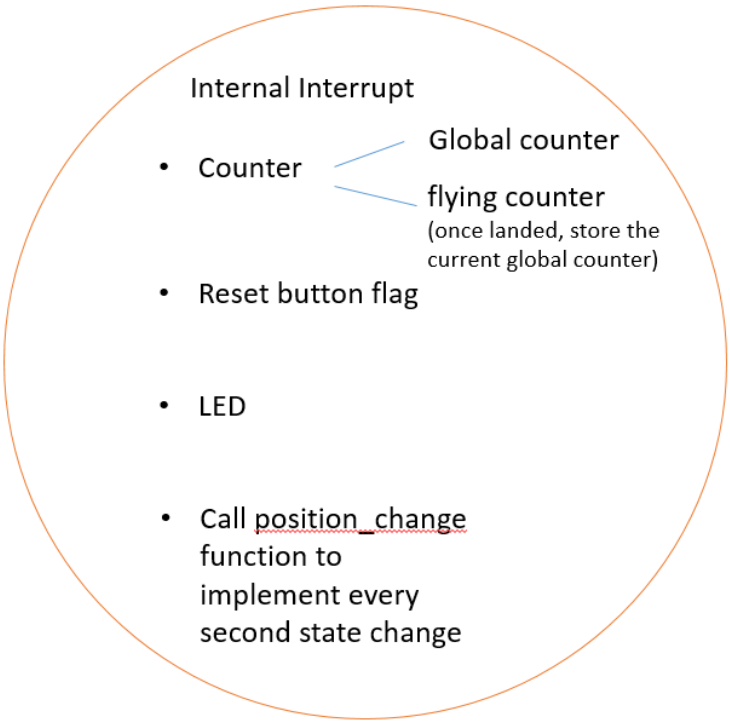
3. Module specification

(1) External Interrupt
a) INT0 is in charge of PB0
b) INT1 is in charge of PB1
In order to guarantee the button could only change the speed once, button flag is required:



c) INT2 is in charge of Motor
The hole_counter is used to store the number of holes. If the hole_counter is 4, the round_counter required to increase one and clear the hole_counter. In this way the motor speed could be control.

(2) Internal Interrupt
Timer0



Internal interrupt have these 4 things to do as the picture shown.

(3) KEYPAD

1(up)	**2(forward)**	**3(down)**
4(left)		6(right)
	8(backward)	
* First press: hover Second press: stop hover	Check hover flag	# First press: take off Second press: landing

Landing state and direction button all need to check hover flag first, because when the helicopter start to hover, other keys cannot affect the flying state. '#' key also has its state_flag to check the helicopter's state.

(4) LED

LED BAR is connected to PORTC. At the beginning, DDRC's value is 0x00.In the design, when the helicopter over boundary, it will call led_bar function that transfer 0xff to DDRC to enable LED works. And the flash control part is in the internal interrupt. To be specific about the control part: every second the LED pattern is the adverse of previous pattern (0x00 and 0xff).

4. Algorithms

(1) Motor speed control:

In this design, there are 5 level speed: 0m/s, 1m/s, 2m/s, 3m/s, 4m/s. In the function named "motor_speed", 4 levels of speed are set according to different value that controls the PWM duty cycle.

(2) Distance test:

The function calculate_distance is used to add the displacement on X-axle and the displacement on Y-axle.

(3) Position change:

This is constituted by a set of functions. Each direction has its own function, for example: in the function of "Forward_change_f", it first check whether the hovering flag is set. If the helicopter is not hovering, it add the speed value to the position X, Y, Z (assume that the forward, right and up are the positive direction). Otherwise, the position do not need to change. After changing the position, it calls a function "flyingState" to display the flying state onto LCD.

Besides, if the next position will over the boundary, it will activate LED.

	Fuction	Subset fuctions
Position_change is called every second (In Timer0OVF).	position_change	FORWARD_change_f
		BACKWARD_change_f
		RIGHT_change_f
		LEFT_change_f
		UP_change_f
		DOWN_change_f

It needs to be emphasized that DOWN_change_f includes the check of the '#' key is whether pressed again (landing), if safely landed it will display the distance and time of flying. Otherwise just as the same as the other direction fuctions.

(4) Display direction:
The function of Display_direction use ASCii code of ‘U’, ’D’, ‘F’, ’B’, ‘L’, ‘R’ to display the directions.

(5) Flying state:
The function of flying_State is used to display the position, direction and speed. Through reading value from the data memory and store value back to the data memory, it could realize the displaying of the flying state.

(6) Hovering state:
The function of hovering_start is used to display the location of the hovering place. And this function is activated by the ‘*’ button.