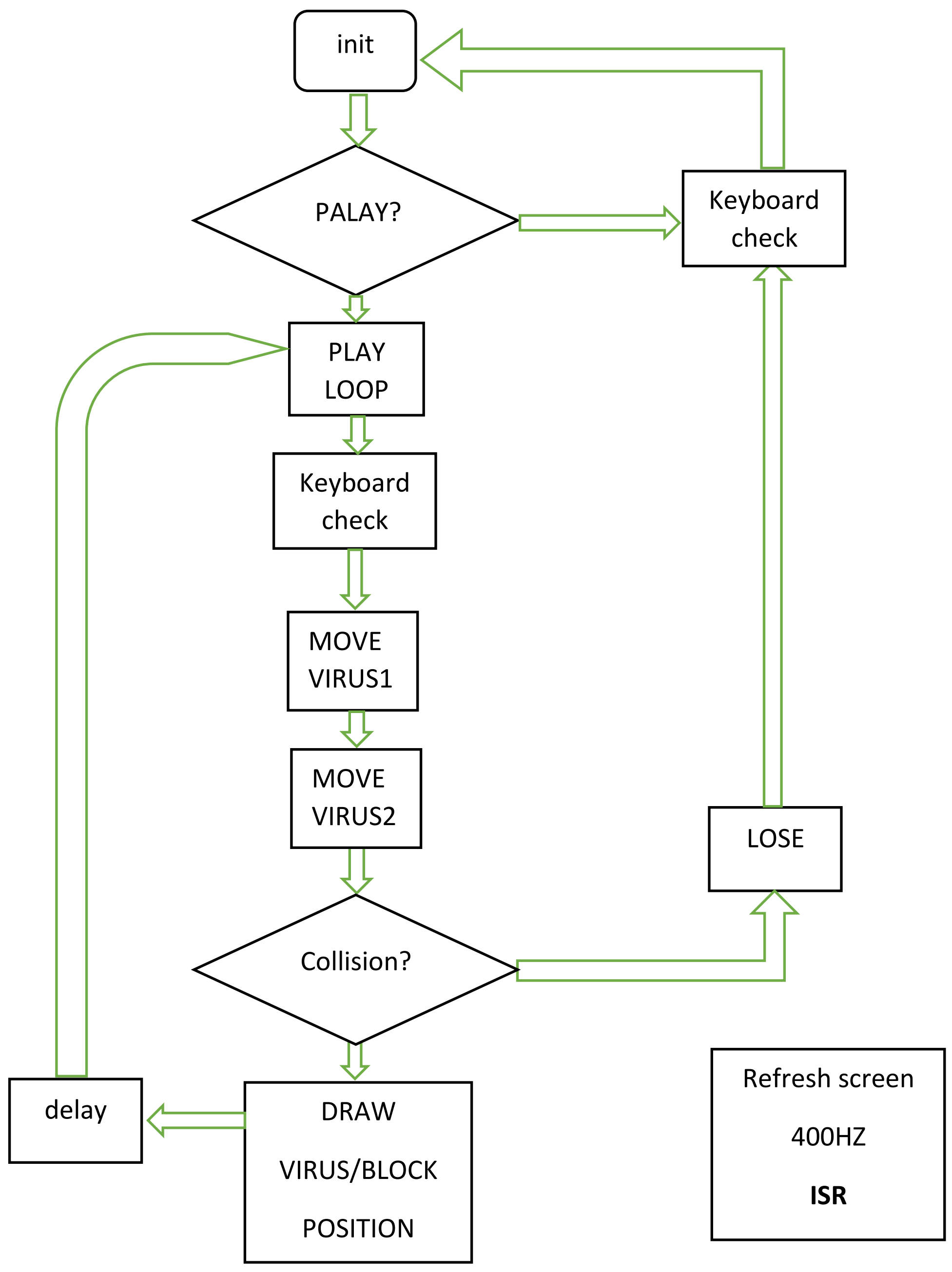
BLOCK SCHEMATIC



Introduction

In this project, we implemented a block Virus Game on ATmega328P micro-controller using AVR Assembly. It can detect the input of keyboard and show changes on the screen.

User Guide

1. **Game cover**: At first you can see a initial page given the name Block virus with pattern of block and the virus, press button 2 to start the game process.
2. **Game process**: At this part, you need to control block movement to avoid conflict with the virus. If you touched the virus during the game, then the sign ”YOU LOSE” will show up.
3. **Restart**: If you want to restart, press button 2 and go back to the initial position to play it again.
4. **Button and boundary**:
   (1) Press button 5,0,1,3 to go up, down, left or right.
   (2) Press button 2 to go to the start position or to restart the game.
   (3) If the user tries to move block out of the screen, it can be detected. block doesn’t move

CODE MAIN IDEA

1. **Screen Display**

   Due to Screen display is independent with the game logic loop, so we put screen display into time interrupt . in other to avoid interrupt affect game loop, we need make sure the screen refresh time should be larger than the running time of game loop , in this project , we choose screen refreshing frequency = 400hz.

   We use timer0 which is 8 bits. The overflow value is 256.

   f_clk = 16MHz / 256 = 62500

   T_CNTinit = 256 – (62500/ 400) = 100

   We divide the screen into 16 blocks each with 7 rows (8 bits in the memory, last bit is set to zero) and 5 columns. From address 0x100 to 0x10F, we store the char buffer for each block. Each char buffer refers to one character defined in the chartable.
   From address 0x10F to 0x100, we read out different value of char buffer and store it in Rx. The address for each character is

   ADR = Chartable + 8*Rx + Row-1.

   First, we get buffer value from the memory. Second, we calculate the address for the character. Third, we get pattern from chartable. Next, we shift out.

2. Button algorithm

We use the 4-steps method to check the button pressing. When a move button is pressed, we read out the position of block through two registers. One stored higher 8 bits, the other stored lower 8 bits.

For the current position, the pattern is set to empty.

Then

(1) sub or add 8 to the low 8 bits register when pressing up or down.

(2) sub or add 1 to the low 8 bits register when pressing left or right.

(3) when new block position is outside the screen, recall Do not move function.

3. Virus movement

we read out the position of virus through two registers. One stored higher 8 bits, the other stored lower 8 bits,

For the current position, the pattern is set to empty.

Then

(1) Sub 1 to low 8 bits register.

(2) when the new virus position is outside the screen, recall jump back function, let virus back to initial position (low 8-bit register add 8)

4. Decision

Under encountering virus, we jump to" LOSE" and then restart.

For instance: When virus low 8-bit register is equal to block virus low 8 bit register