



ផែបាតីម៉ង់: ពតិមានវិទ្យា

Subject : Computer Architecture

Project : Arduino Game

បង្ក្រោនដោយលោកគ្រូ : Ouk Polyvann

ក្រុមទី ១

ល.រ	ឈ្មោះ	ពិន្ទុ	គ្នានាគី	ភាគរយនៃការយល់ដឹង	មេរោន 5, 8
១	ស្តី ចាន់វូដែម		Code, Slide, Simulator, hardware	90%	90%
២	សាំង មិញ្ញសី			50%	50%
៣	វិន ម៉ែងទ្វួន			50%	50%
៤	លី គុលា			50%	50%
៥	វិបុល សុខលីម			50%	50%
៦	លី ម៉ែងហោង		ផ្លូវតាមប័ណ្ណ	70%	70%
៧	ស៊ុន ចាន់ផែ ហុង			50%	50%
៨	វ៉ាង វាន់ចេន			50%	50%

INTRODUCTION

នៅពេលបច្ចុប្បន្ន យើងយើងឱ្យបាបច្រេកវិញរួចរាល់កំពុងមានការវិភាគមើនយោងខ្សោះ គោលនយកបច្រេកវិញនេះទៅបំពាក់លើយន្តដើម្បីអាយរចយន្តមានភាពឆ្លាត់នៃអាជីវការជាយករារបញ្ហារតាម គេឡើ តាមទូរស្សពីជាជីម ដើម្បីសម្រេចដល់ការប្រើប្រាស់នឹងមានភាពជាយករារបញ្ហារតាមទូរស្សពីជាជីម គ្រប់រិស៊ីយៈ

- វិស៊ីយកសិកម្ម
- វិស៊ីយឧស្សាហកម្ម
- វិស៊ីយអប់រំ
- វិស៊ីយយោធ



Arduino LCD Run Game

Arduino LCD Game require:

-Arduino Board



-Momentary button or Switch



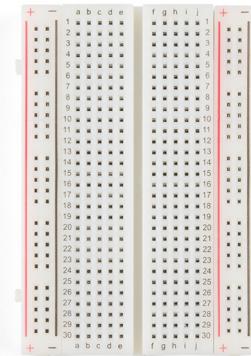
-10K ohm resistor



-hook-up wires



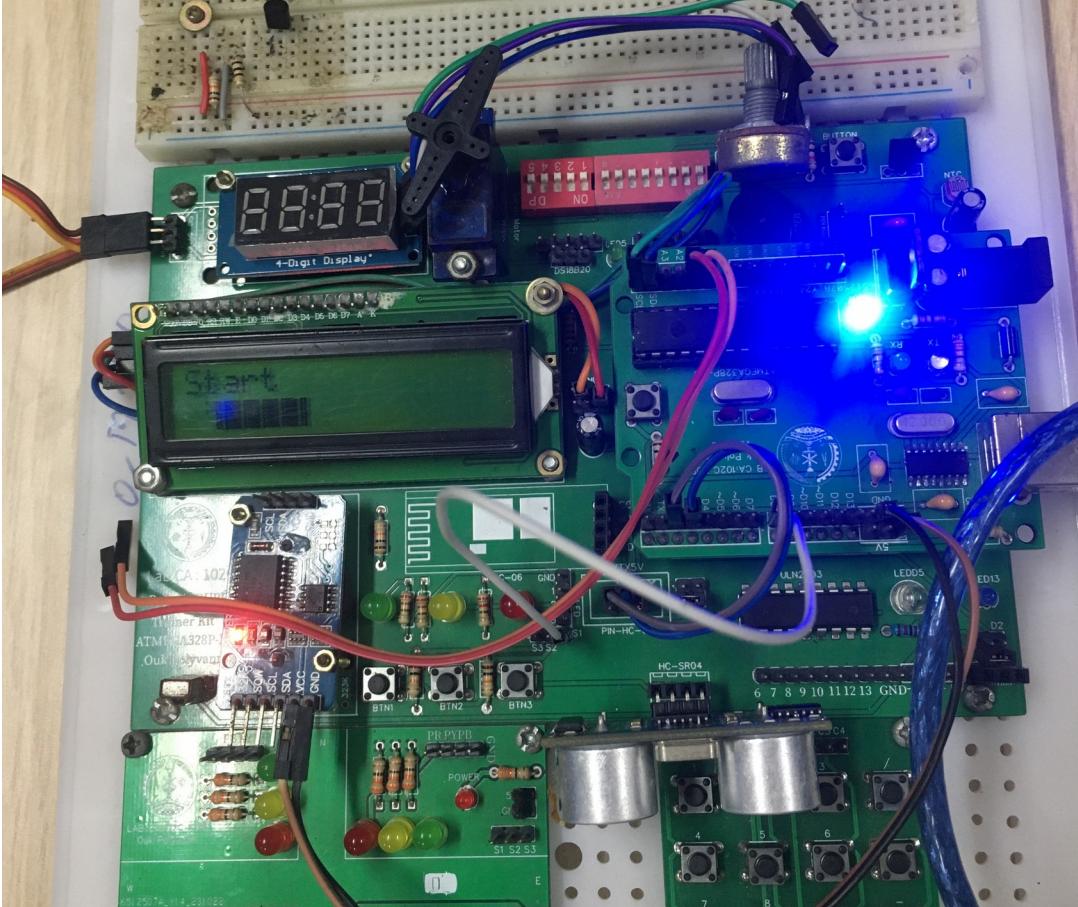
-breadboard



-LCD

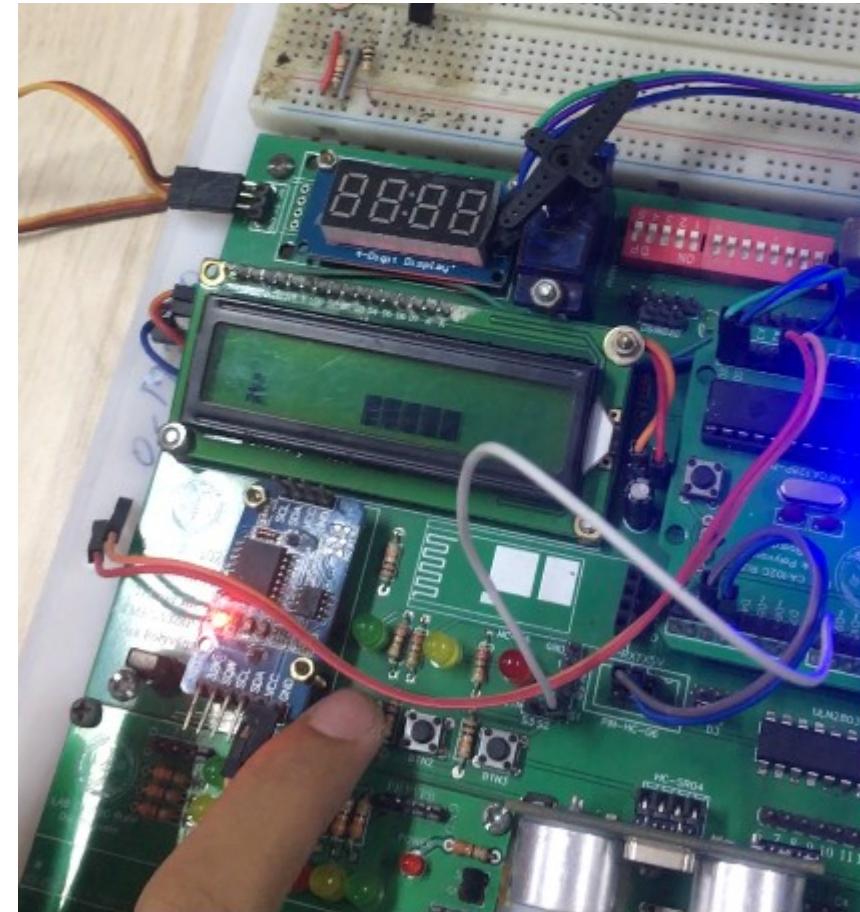
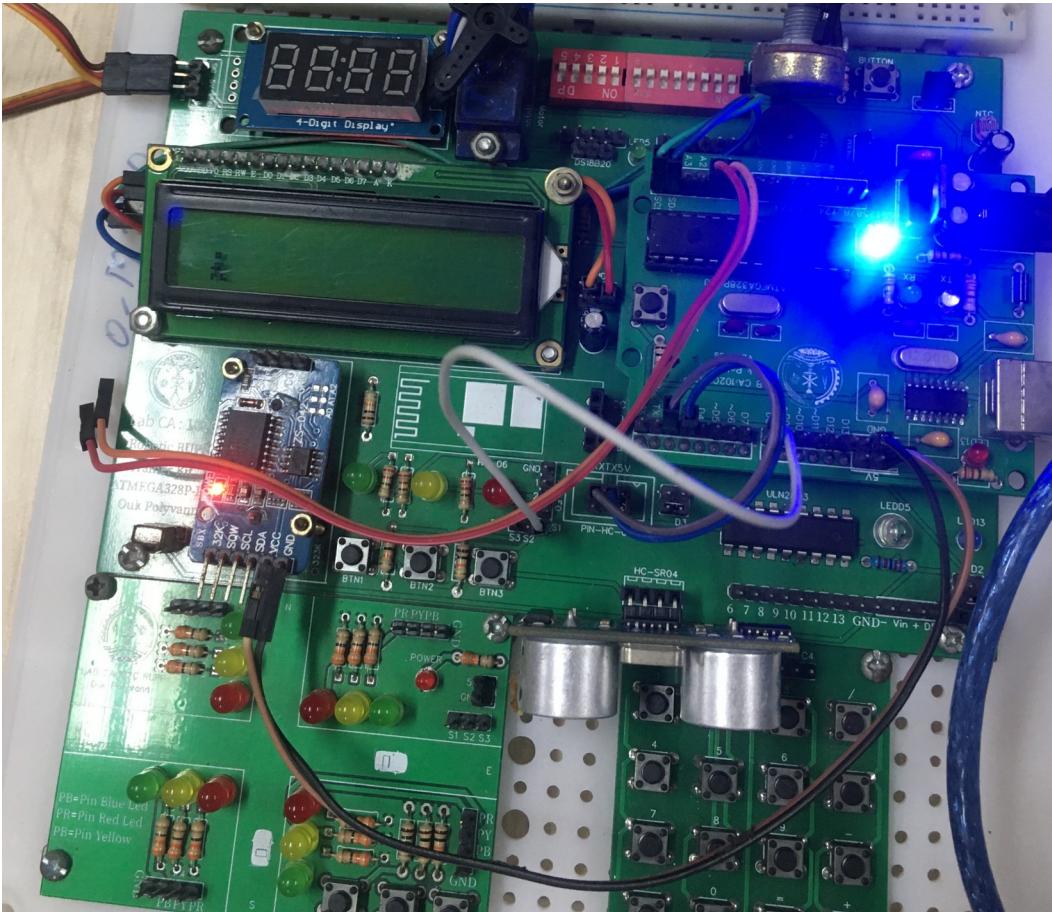


ណែនាំអំពី GAME

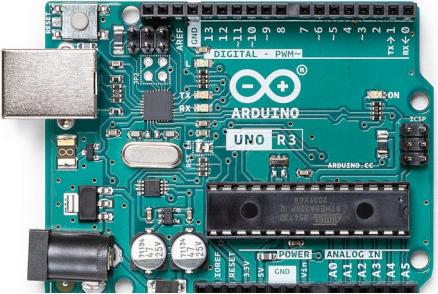


- ◆ Arduino LCD Run Game គីជារណែនាំដែលតម្រូវការលោតផ្ទាន់របង់។
- ◆ ដើម្បីលើងGameមួយនេះអ្នកត្រូវចូចButton
- ◆ កាលណាមួកបែនបែននៅ៖ ត្រូវបានចុះថាថ្វូរ ហើយត្រូវបានបញ្ចប់។
- ◆ ចូចButtonជាប់គីជារលោត ប្រើបានគីជារដូចមកវិញ
- ◆ អ្នកអាចលើងវាសារជាថីបានដោយគ្រាន់តែចូចButton នៅ៖ Gameនឹងចាប់ផ្តើមមួងឡើត។

រូបភាពដំណើរការនៃ GAME



Hardware



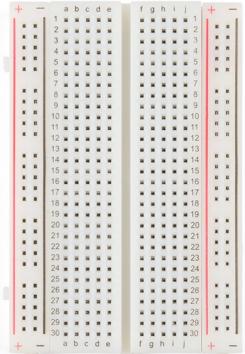
Arduino Uno គឺជាបន្ទះមីក្រត្រិក្សប្រភពបើកចំហែងមានមូលដ្ឋានលើ Microchip ATmega328P microcontroller (MCU) និងត្រូវបានបង្កើតឡើងដោយ Arduino.cc ហើយត្រូវបានចេញផ្សាយដំបូងក្នុងឆ្នាំ 2010។



LCD 16x2 គឺជាប្រភេទអេក្រង់ត្រូវស្ថាល់រវ ដែលមានបង្ហាញដល់ខ្លះ 16តួអក្សរក្នុងមួយបន្ទាត់ និង 2បន្ទាត់។ ការបង្ហាញទាំងនេះត្រូវបានប្រើប្រាស់យ៉ាងខ្ពស់នៅក្នុងកម្មវិធី ដោយបង្កើតឡើងដំបូងក្នុងកម្មវិធី ដើម្បីបង្ហាញព័ត៌មានអត្ថបទ បុទ្ធន៍យនោក្នុងកម្មវិធី។



បូតុងរូប បុបូតុងសាមញ្ញ គឺជាយនការប្រដែលសាមញ្ញមួយ ដើម្បីគ្រប់គ្រងទិន្នន័យចំណុចនៃម៉ាសីន បុណ្ណោះនៅក្នុងបូតុងជាគម្ពាលប្រព័ន្ធដែលបានផលិតចេញពីរត្តិវិធី ជាគម្ពាលប្រាសិក បុណ្ណោះ



breadboard គឺជាមូលដ្ឋានសំណង់ដែលប្រព័ន្ធដែលប្រើប្រាស់សាងគំរូណាក់កណ្តាលអចិន្តូយ៉ាន់
សេវ្យគីឡូចត្រូនិ៍ មិនដូច perfboard ឬ stripboard ទេ breadboards មិនតម្រូវឱ្យមានការ
soldering បុរាណបំផ្តាញផ្លូវហើយដូចតែគីឡូចត្រូនិ៍ សម្រាប់ហេតុផលនេះ ភាពខ្សោនក៏
មានប្រជាប្រើប្រាស់ជាមួយសិស្សនិស្ស និងកូងការអប់រំដោយបច្ចេកវិទ្យាផងដោ។



Hook-Up Wire គឺជាដែូនដែលមានអីសុទ្ធផ្លែងតែមួយ ដែលអាចប្រើសម្រាប់កម្មវិធីដែលមាន
តាមស្រួលទាប និងចរន្តទាប។ ខ្សោយ Hook-Up ដែលបានយ៉ាងល្អនៅក្នុងកន្លែងបិទិត
ហើយភ្លាប់មកជាមួយនូវសំណងជាថ្មីនប្រចាំនបាយទីក អីសុទ្ធផ្លែង។

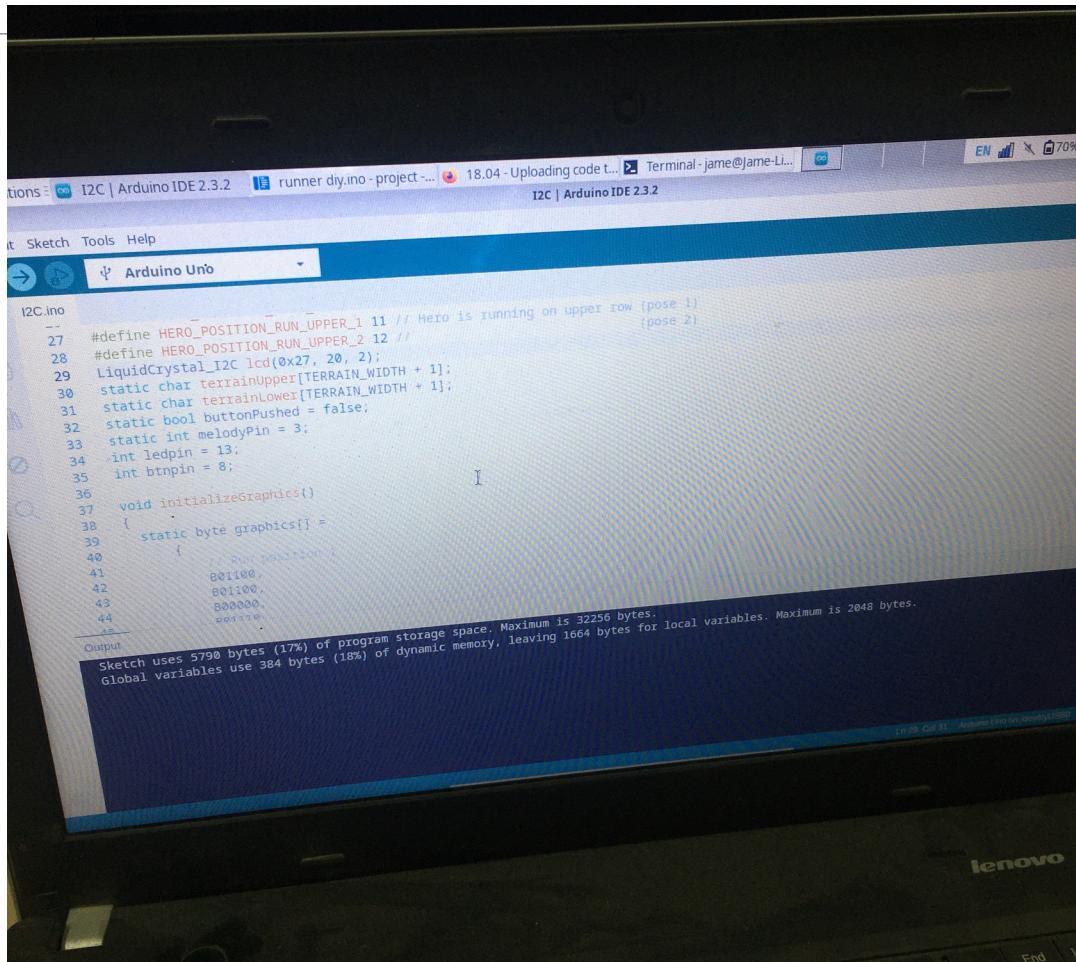
ការតាម្លៃងArduino Run Game

- ត្រូវបានArduino Board ទៅកាន់Lcd(A5->SCL, A4 → SDA,GND->ground,VCC->+5V)
- ត្រូវបានArduino Board ទៅកាន់Button(PIN8, 5V)

CODE

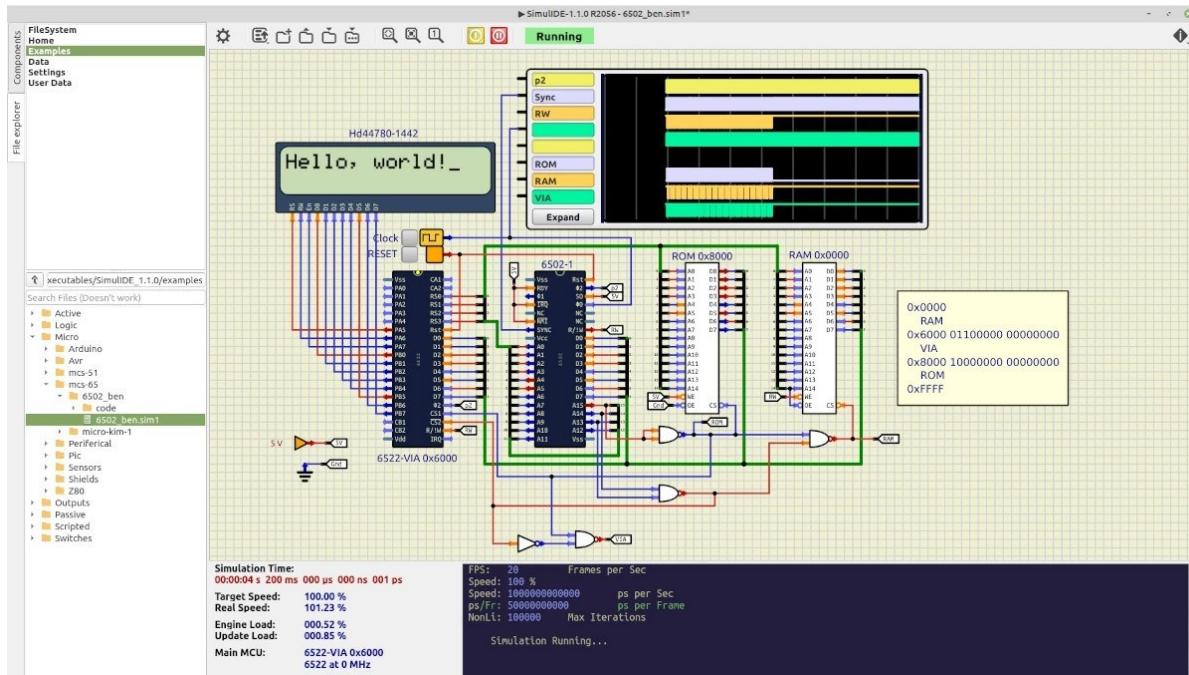
```
1 #include <LiquidCrystal_I2C.h>
2 #define SPRITE_RUN1 1
3 #define SPRITE_RUN2 2
4 #define SPRITE_JUMP 3
5 #define SPRITE_JUMP_UPPER '.' // Use the '.' character for the head
6 #define SPRITE_JUMP_LOWER 4
7 #define SPRITE_TERRAIN_EMPTY '' // Use the '' character
8 #define SPRITE_TERRAIN_SOLID 5
9 #define SPRITE_TERRAIN_SOLID_RIGHT 6
10 #define SPRITE_TERRAIN_SOLID_LEFT 7
11 #define HERO_HORIZONTAL_POSITION 1 // Horizontal position of hero on screen
12 #define TERRAIN_WIDTH 16
13 #define TERRAIN_EMPTY 0
14 #define TERRAIN_LOWER_BLOCK 1
15 #define TERRAIN_UPPER_BLOCK 2
16 #define HERO_POSITION_OFF 0 // Hero is invisible
17 #define HERO_POSITION_RUN_LOWER_1 1 // Hero is running on lower row (pose 1)
18 #define HERO_POSITION_RUN_LOWER_2 2 // (pose 2)
19 #define HERO_POSITION_JUMP_1 3 // Starting a jump
20 #define HERO_POSITION_JUMP_2 4 // Half-way up
21 #define HERO_POSITION_JUMP_3 5 // Jump is on upper row
22 #define HERO_POSITION_JUMP_4 6 // Jump is on upper row
23 #define HERO_POSITION_JUMP_5 7 // Jump is on upper row
24 #define HERO_POSITION_JUMP_6 8 // Jump is on upper row
25 #define HERO_POSITION_JUMP_7 9 // Half-way down
26 #define HERO_POSITION_JUMP_8 10 // About to land
27 #define HERO_POSITION_RUN_UPPER_1 11 // Hero is running on upper row (pose 1)
28 #define HERO_POSITION_RUN_UPPER_2 12 // (pose 2)
29 // LiquidCrystal_I2C lcd(80, 16, 2);
30 LiquidCrystal_I2C lcd(0x27, 16, 2);
31 static char terrainUpper[TERRAIN_WIDTH + 1];
32 static char terrainLower[TERRAIN_WIDTH + 1];
33 static bool buttonPushed = false;
34 static int melodyPin = 3;
35 int ledpin = 13;
36 int bttnpin = 8;
37
38 void initializeGraphics()
39 {
40     static byte graphics[] =
41     {
42         // Run position 1
43         B01100,
44         B01100,
45         B00000,
46         B01110,
47         B11100,
48         B01100,
49         B11010,
50         B10011,
51         // Run position 2
52         B01100,
53         B01100,
54         B00000,
55         B01100,
56         B01100,
57         B01100,
58         B01100,
59         B01110,
60         // Jump
```

Code Arduino Run game ເນື່ງເລີ້ມ Arduino IDE



```
Applications runner diy.ino - arduino - ... Arduino Game.odp — Lib... sketch_may09a | Arduin... zip - Th sketch_may09a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_may09a S
#include <LiquidCrystal_I2C.h>
#define SPRITE_RUN1 1
#define SPRITE_RUN2 2
#define SPRITE_JUMP 3
#define SPRITE_JUMP_UPPER '.' // Use the '.' character for the head
#define SPRITE_JUMP_LOWER 4
#define SPRITE_TERRAIN_EMPTY '' // Use the '' character
#define SPRITE_TERRAIN_SOLID 5
#define SPRITE_TERRAIN_SOLID_RIGHT 6
#define SPRITE_TERRAIN_SOLID_LEFT 7
#define HERO_HORIZONTAL_POSITION 1 // Horizontal position of hero on screen
#define TERRAIN_WIDTH 16
#define TERRAIN_EMPTY 0
#define TERRAIN_LOWER_BLOCK 1
#define TERRAIN_UPPER_BLOCK 2
#define HERO_POSITION_OFF 0 // Hero is invisible
#define HERO_POSITION_RUN_LOWER_1 1 // Hero is running on lower row (pose 1)
#define HERO_POSITION_RUN_LOWER_2 2 // (pose 2)
#define HERO_POSITION_JUMP_1 3 // Starting a jump
#define HERO_POSITION_JUMP_2 4 // Half-way up
#define HERO_POSITION_JUMP_3 5 // Jump is on upper row
#define HERO_POSITION_JUMP_4 6 // Jump is on upper row
#define HERO_POSITION_JUMP_5 7 // Jump is on upper row
#define HERO_POSITION_JUMP_6 8 // Jump is on upper row
#define HERO_POSITION_JUMP_7 9 // Half-way down
#define HERO_POSITION_JUMP_8 10 // About to land
#define HERO_POSITION_RUN_UPPER_1 11 // Hero is running on upper row (pose 1)
#define HERO_POSITION_RUN_UPPER_2 12 // (pose 2)
LiquidCrystal_I2C lcd(80, 16, 2);
```

Simulation



- SimulIDE is a simple real time electronic circuit simulator, intended for hobbyist or students to learn and experiment with analog and digital electronic circuits and microcontrollers.
- It supports PIC, AVR , Arduino and other MCUs and MPUs
- Thus, We use SimulIDE to simulate our ARDUINO LCD Game.

Circuit of Arduino LCD GAME

The screenshot shows a breadboard setup in SimulIDE. On the left, a breadboard with a blue Arduino Uno and a green HD44780-36 LCD module is shown. The Uno pins are labeled with their functions. The LCD module is connected to the Uno's I2C pins (SDA, SCL), digital pins (D9-D13), and an interrupt pin (INT). A push button is connected between GND and digital pin 13. On the right, the SimulIDE interface is visible, including a component library browser, a schematic editor window showing the same setup, and a code editor window displaying the C++ source code for the LCD game.

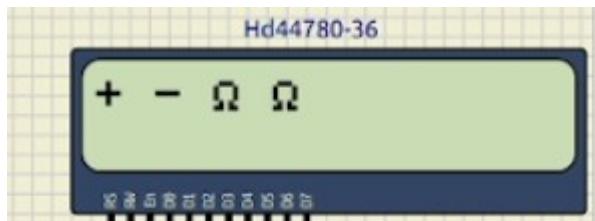
Code Editor (runner diy.ino):

```
1 #include <LiquidCrystal_I2C.h>
2 #define SPRITE_RUN1 1
3 #define SPRITE_RUN2 2
4 #define SPRITE_JUMP 3
5 #define SPRITE_JUMP_UPPER '.' // Use the '.' character for the head
6 #define SPRITE_JUMP_LOWER 4
7 #define SPRITE_TERRAIN_EMPTY '' // Use the '' character
8 #define SPRITE_TERRAIN_SOLID 5
9 #define SPRITE_TERRAIN_SOLID_RIGHT 6
10 #define SPRITE_TERRAIN_SOLID_LEFT 7
11 #define HERO_HORIZONTAL_POSITION 1 // Horizontal position of hero on screen
12 #define TERRAIN_WIDTH 16
13 #define TERRAIN_EMPTY 0
14 #define TERRAIN_LOWER_BLOCK 1
15 #define TERRAIN_UPPER_BLOCK 2
16 #define HERO_POSITION_OFF 0 // Hero is invisible
17 #define HERO_POSITION_RUN_LOWER_1 1 // Hero is running on lower row (pose 1)
18 #define HERO_POSITION_RUN_LOWER_2 2 // (pose 2)
19 #define HERO_POSITION_JUMP_1 3 // Starting a jump
20 #define HERO_POSITION_JUMP_2 4 // Half-way up
21 #define HERO_POSITION_JUMP_3 5 // Jump is on upper row
22 #define HERO_POSITION_JUMP_4 6 // Jump is on upper row
23 #define HERO_POSITION_JUMP_5 7 // Jump is on upper row
24 #define HERO_POSITION_JUMP_6 8 // Jump is on upper row
25 #define HERO_POSITION_JUMP_7 9 // Half-way down
26 #define HERO_POSITION_JUMP_8 10 // About to land
27 #define HERO_POSITION_RUN_UPPER_1 11 // Hero is running on upper row (pose 1)
28 #define HERO_POSITION_RUN_UPPER_2 12 // (pose 2)
29 LiquidCrystal_I2C lcd(80, 16, 2);
30 // LiquidCrystal_I2C lcd(0x27, 16, 2);
31 static char terrainUpper[TERRAIN_WIDTH + 1];
32 static char terrainLower[TERRAIN_WIDTH + 1];
33 static bool buttonPressed = false;
```

Firmware Uploaded to 1@mega328-109(mega328)
/home/jane/Desktop/project/arduino/SimulIDE/runner diy.runner diy.hex

Searching for variables... 45 variables found
Searching for Functions... 33 functions found
Mapping Flash to Source... 466 lines mapped

Arduino Print Game



- ◆ Arduino Print Game គឺជារៀង់ចាយដែលអាចទូនាសក្រាមានភាពសំរើកភ្លើងចងចាំ
- ◆ Game នេះតម្រូវអ្នកផ្តើសវិសយករូបណាមួយនៅលើ LCD
- ◆ បើអ្នកវិសយករូបណាដែលអ្នកបានវិសរួចហើយនោះ you lose the game
- ◆ ដើម្បីលេង game នេះអ្នកគ្រាន់តែចុបលើ But ton ណាមួយកំបាន
- ◆ អ្នកផ្តើសវិសរូបភាពភាមលំដាប់លំដោយនៃ Button

ការតម្លៃងArduino Print Game

- ត្រូវArduino Board ទៅកាន់Lcd(A5->SCL, A4->SDA, GND->ground,VCC->+5V)
- ត្រូវArduino Board ទៅកាន់ 4 Button(PIN6,PIN7,PIN8,PIN9,5V)
- ការតម្លៃងមានភាពស្រស់ដៃងទៅនឹងArduino Run Game

Code Arduino Print Game

```
/* Arduino print game
written by brorojame2030@gmail.com */
#include <LiquidCrystal_I2C.h>
#include <StandardCplusplus.h>
#include <vector>
#include <ctime>
#include <cstdlib>
#include "img.h"
LiquidCrystal_I2C lcd(80, 16, 2);
std::vector<byte *> img;
const short btn1 = 6;
const short btn2 = 7;
const short btn3 = 8;
const short btn4 = 9;
void push()
{
    img.push_back(plus);
    img.push_back(minus);
    img.push_back(Heart);
    img.push_back(Bell);
    img.push_back(human);
    img.push_back(smileyFace);
    img.push_back(omega);
    img.push_back(arrows);
}
void setup()
{
    Serial.begin(9600);
    lcd.init();
    lcd.backlight();
    /*     byte* img[] = {plus,minus};
        lcd.createChar(0, img[0]); */
}
void start()
{
    std::vector<int> answered;
    bool lost = false;
    for (; !lost;)
    {
        std::vector<int> img_printed;
        lcd.clear();
        for (int i : {0, 2, 4, 6})
        {
            lcd.setCursor(i, 0);
            int r = random(0, 7);
            for (int e : img_printed)
            {
                if (e == r)
                {
                    // r = random(0, 7);
                    r = (r + random(0, 7)) / 2;
                }
            }
            delay(1000);
            img_printed.push_back(r);
            lcd.write(r);
        }
        int tmp;
        while (1)
        {
            if (btnstate1 == HIGH)
            {
                tmp = img_printed.at(0);
                break;
            }
            else if (btnstate2 == HIGH)
            {
                tmp = img_printed.at(1);
                break;
            }
            else if (btnstate3 == HIGH)
            {
                tmp = img_printed.at(2);
                break;
            }
            else if (btnstate4 == HIGH)
            {
                tmp = img_printed.at(3);
                break;
            }
        }
        for (int a : answered)
        {
            if (tmp == a)
            {
                lcd.clear();
                lcd.print("you lost");
                lost = true;
            }
        }
        answered.push_back(tmp);
        if (answered.size() == 7)
        {
            lcd.clear();
            lcd.print("you win");
            break;
        }
    }
}
```

Circuit and Simulation of Arduino Print Game

