ДОДАТОК А
ГРАФІЧНІ МАТЕРІАЛИ

ДОДАТОК Б
КОД СКЕТЧУ

HoloDisplay.ino

```
#define NUM_LEDS 72
#define BRIGHTNESS 200
#define MOTOR_MAX 180
#define RES 10
#define COEF 0.3
#define PODGON 1.3
#define OFFSET 240
#define NUM_FRAMES 8
#define FRAME_RATE 60

#define PIN 4
#define MOS 3
#define BTN1 8
#define BTN1_G 6


//#include "homer.h"
#include "sonic.h"
//#include "sponge.h"


#include <TimerOne.h>
#include "Adafruit_NeoPixel.h"

Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_LEDS, PIN, NEO_GRB +
NEO_KHZ800);
#include "gamma.h"

boolean motor_state = false;
volatile uint32_t timer;
volatile uint32_t period, new_period, period_f;
volatile boolean timer_isr, hall_isr, new_loop;

uint16_t counter1, counter2;
uint8_t counter;
const byte n_segm = 360 / RES;
const byte offset = (float)OFFSET / 360 * n_segm;
//const byte offset = 0;
byte this_frame = 0, count_frame;

static uint32_t expandColor(uint16_t color) {
  return ((uint32_t)pgm_read_byte(&gamma5[ color >> 11       ]) << 16) |
         ((uint32_t)pgm_read_byte(&gamma6[(color >> 5) & 0x3F]) <<  8) |
         pgm_read_byte(&gamma5[ color       & 0x1F]);
}

void setup() {
  Serial.begin(115200);
  strip.begin();
  strip.setBrightness(BRIGHTNESS);

  pinMode(BTN1, INPUT_PULLUP);
  pinMode(BTN2, INPUT_PULLUP);
  pinMode(BTN1_G, OUTPUT);
  pinMode(BTN2_G, OUTPUT);
```

```
    digitalWrite(BTN1_G, LOW);
    digitalWrite(BTN2_G, LOW);
    pinMode(MOS, OUTPUT);

    attachInterrupt(0, hall, RISING);
    Timer1.initialize();
    Timer1.stop();

    strip.clear();
    strip.show();
  }

  void hall() {
    if (micros() - timer > 30000) {
      new_period = micros() - timer;
      timer = micros();
      hall_isr = true;
    }
  }

  void timer_interrupt() {
    timer_isr = true;
  }

  void setPix(byte i, int counter, int this_pix) {
    switch (this_frame) {
      case 0: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame1[counter][this_pix]))));
        break;
      case 1: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame1[counter][this_pix]))));
        break;
      case 2: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame2[counter][this_pix]))));
        break;
      case 3: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame3[counter][this_pix]))));
        break;
      case 4: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame4[counter][this_pix]))));
        break;
      case 5: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame5[counter][this_pix]))));
        break;
      case 6: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame6[counter][this_pix]))));
        break;
      case 7: strip.setPixelColor(i,
expandColor(pgm_read_word(&(frame7[counter][this_pix]))));
        break;
    }
  }

  void animation() {
    timer_isr = false;
    strip.clear();

    byte this_pix = 0;
    for (int i = NUM_LEDS / 2 - 1; i >= 0; i--) {
      setPix(i, counter1, this_pix);
      this_pix++;
    }
    this_pix = 0;
    for (int i = NUM_LEDS / 2 ; i < NUM_LEDS; i++) {
```

```
      setPix(i, counter2, this_pix);
      this_pix++;
    }

    counter1++;
    counter2++;
    if (counter2 > n_segm) counter2 = 0;
    if (counter1 > n_segm) counter1 = 0;

    strip.show();
  }

  void loop() {
    if (hall_isr) {
      hall_isr = false;
      counter1 = 0 + offset;
      counter2 = n_segm / 2 + offset;
      if (counter2 > n_segm) counter2 = counter2 - n_segm;

      count_frame++;
      if (count_frame >= FRAME_RATE) {
        count_frame = 0;
        this_frame = ++this_frame % NUM_FRAMES;
      }

      animation();
      period_f = (float)new_period * COEF + (float)period_f * ((float)1 -
COEF);
      Timer1.setPeriod(period_f / n_segm * PODGON);
      Timer1.restart();
      //Serial.println(isr_counter2);
    }

    if (timer_isr) {
      timer_isr = false;
      animation();
    }

    if (!digitalRead(BTN1)) {
      motor_state = !motor_state;
      if (motor_state) {
        for (int i = 0; i < MOTOR_MAX; i++) {
          analogWrite(MOS, i);
          delay(20);
        }
        Timer1.attachInterrupt(timer_interrupt);
      } else {
        for (int i = MOTOR_MAX; i > 0; i--) {
          analogWrite(MOS, i);
          delay(20);
        }
        analogWrite(MOS, 0);
        Timer1.stop();
        Timer1.detachInterrupt();
        delay(500);
        timer_isr = false;
        strip.clear();
        strip.show();
      }
    }
  }
```

Sonic.h

```
 const uint16_t frame0[][36] PROGMEM={
 {0xE4EF,  0xE510,  0xED10,  0x9AC7,  0x4928,  0x5ADA,  0x631D,  0x6B5D,  0x52FD,
0x39F4,  0x9493,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //10 + 0
 {0xE4EF,  0xE510,  0xE510,  0x416A,  0x21F8,  0x529A,  0x631C,  0x631D,  0x631D,
0x4A79,  0x4A25,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //20 + 0
 {0xE510,  0xE510,  0xC3A9,  0x18F1,  0x2114,  0x4A79,  0x4215,  0x633D,  0x5AFC,
0x31EF,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //30 + 0
 {0xE510,  0xE510,  0x390B,  0x2115,  0x2239,  0x527A,  0x0807,  0x631D,  0x4218,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //40 + 0
 {0xE532,  0xFD2F,  0x1048,  0x20D4,  0x2218,  0x5AFD,  0x212F,  0x633D,  0x4238,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //50 + 0
 {0xE532,  0x9C93,  0x18AA,  0x2114,  0x19F8,  0x7BB9,  0x3A4E,  0x633D,  0x4A7A,
0xCE39,  0xE75C,  0x5BAB,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //60 + 0
 {0xCDB7,  0x8C94,  0x4A50,  0x2114,  0x6BD9,  0xD69C,  0xFFFF,  0xFFFF,  0xFFFF,
0xDEDC,  0xB5B7,  0xE73D,  0x0841,  0x0000,  0x0000,  0x0000,  0x0000,  0x0000,
0x0000,  0x0000,  0xF0E4,  0xEE76,  0xE082,  0xE082,  0xE082,  0x3820,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //70 + 0
 {0xCDB7,  0xBDF8,  0x8413,  0x526C,  0xB576,  0xC619,  0xC639,  0xDEFC,  0xFFFF,
0xFFFF,  0xC63A,  0xC4B3,  0xE021,  0xE082,  0xEAEB,  0xFFFF,  0xEA08,  0xE082,
0xE082,  0xE082,  0xE082,  0xE082,  0xE082,  0xD862,  0x4020,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //80 + 0
 {0xBDF9,  0xE73D,  0xF77F,  0xCE5A,  0x31A6,  0x94B4,  0xC5F9,  0xC5F9,  0xC619,
0xD1E7,  0xE082,  0xE082,  0xE082,  0xE082,  0xE082,  0xFFFB,  0xAC60,  0xFF20,
0xE802,  0xD082,  0x3020,  0x6841,  0x6041,  0x5841,  0x0000,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //90 + 0
 {0xC5F8,  0xE71D,  0xE73D,  0xEF3E,  0xCE5A,  0x4A49,  0x9CD5,  0x6820,  0xA861,
0xA061,  0xA061,  0xA061,  0xA061,  0xA061,  0xA061,  0xA861,  0x4000,  0x3965,
0x8B0E,  0x94B4,  0x94B4,  0x9CF6,  0x7ACC,  0x834D,  0x2103,  0x0000,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //100 + 0
 {0xC5F8,  0xEF5E,  0xE73E,  0xE73E,  0xEF3E,  0xC5F9,  0x2148,  0x4020,  0x6041,
0x7841,  0x9020,  0x9041,  0x7020,  0x5040,  0x738E,  0x94B4,  0x94B4,  0x94B4,
0x6B4D,  0x6020,  0x6041,  0x6041,  0x6041,  0x6041,  0x6041,  0x2103,  0x0000,
0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,    0x0000,
0x0000 },   //110 + 0
 };
```

№	Позначення	Найменування	Дод. відомості
		Текстові документи	
1	ГЮІК. 46741Х.021 ПЗ	Пояснювальна записка	69 с.
		Графічні документи	
2		Слайд-презентація	13 слайдів
		Носії інформації	
3		Компакт-диск (CD)	1 шт.

Змін.	Арк.	Номер докум.	Підп.	Дата	ГЮІК. 46741Х.021 ВД			
Розроб.		Шапошник К.А.			Голографічний проектор на базі платформи Arduino Nano	Літ	Аркуш	Аркушів
Перевір.		Чумаченко С.В.				У		
Н.контр.		Шкіль О.С.			Відомість кваліфікаційної роботи	ХНУРЕ Кафедра АПОТ		
Затв.		Чумаченко С.В.						