МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА ИНДИВИДУАЛЬНОГО ПРОЕКТА ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ ВСТРОЕННЫХ СИСТЕМ»**

Выполнила:

Студентка 2 курса 1 группы

Угоренко В.Р.

Проверил:

Старший преподаватель

Королёв А.А.

Минск 2024

**Содержание**

[Введение 3](#_Toc186067040)

[Описание проекта 4](#_Toc186067041)

[Заключение 6](#_Toc186067042)

[Схема подключения 7](#_Toc186067043)

[Листинг кода 7](#_Toc186067044)

# **Введение**

В современном мире люди всё чаще стремятся внедрять технологии в повседневную жизнь, делая её более комфортной, увлекательной и функциональной. Одним из ярких примеров таких решений является проект "Растение-тамагочи", который представляет собой интерактивный горшок для растения. Это устройство не только помогает заботиться о растении, но и делает процесс ухода эмоционально насыщенным и привлекательным.

Идея проекта заключается в создании умного устройства, которое реагирует на действия владельца: полив, прикосновения и изменение освещения. При этом устройство отображает эмоциональное состояние растения с помощью OLED-дисплея, что позволяет пользователю чувствовать обратную связь. Такой подход не только добавляет игровой элемент в уход за растением, но и способствует формированию более ответственного отношения к окружающей среде.

Проект "Растение-тамагочи" разработан на базе Arduino UNO, которая предоставляет широкие возможности для работы с различными сенсорами и компонентами. Использование Arduino делает проект доступным для реализации даже начинающими инженерами и студентами.

Этот проект может быть полезен не только для домашних пользователей, но и для образовательных целей. Он демонстрирует, как встроенные системы и простые электронные компоненты могут быть использованы для решения творческих задач и разработки уникальных устройств. "Растение-тамагочи" является отличным примером того, как технологии могут сделать уход за растением более интересным и осмысленным.

# **Описание проекта**

Данный проект — это интерактивный горшок для растения, работающий на базе Arduino UNO. Он использует датчики для определения уровня влажности почвы, освещённости и прикосновений. OLED-дисплей отображает текущее состояние растения.

*Основные компоненты проекта:*

1. Arduino UNO — платформа, обеспечивающая управление всей системой.
2. Датчик влажности почвы YL-38 — определяет уровень увлажнённости почвы.
3. OLED-дисплей IIC 0.91" SSD1306 — выводит анимации и сообщения о состоянии растения.
4. Датчик света XD-80 — определяет уровень освещения.
5. Сенсорный ёмкостный датчик EM-502 — регистрирует прикосновения к горшку.

**Принцип работы:**

*Реакция на прикосновения*

Когда пользователь проводит рукой по сенсорному датчику, устройство регистрирует сигнал высокого уровня (High) от сенсора. Этот сигнал воспринимается Arduino как команда "погладить" растение. В ответ на это растение "радуется", и на OLED-дисплее отображается анимация, символизирующая радость.

Сенсорный датчик позволяет создать иллюзию взаимодействия с живым существом. Важным моментом является чувствительность сенсора: он реагирует только на прямое касание, исключая случайные срабатывания от других объектов. Реакция активируется мгновенно, обеспечивая плавность и реалистичность "общения" с растением.

*Реакция на полив*

Контроль влажности почвы осуществляется с помощью датчика YL-38. Этот датчик измеряет уровень увлажнённости и передаёт данные на Arduino. Устройство проверяет уровень влажности почвы не постоянно, а с заданным интервалом — раз в час. Такой подход предотвращает повторное воспроизведение реакции на недавний полив и позволяет корректно определять момент, когда почва действительно становится сухой.

Когда пользователь поливает растение, датчик регистрирует изменение влажности и передаёт сигнал, который интерпретируется Arduino как факт полива. В ответ на дисплее появляется сообщение благодарности. Эта задержка между проверками была добавлена специально, чтобы исключить возможность ложных срабатываний и дать растению "отдохнуть" от избыточных действий.

Для реализации этого функционала используется таймер, встроенный в программу Arduino. Он отсчитывает время с последней проверки состояния влажности почвы. Если с момента предыдущей проверки прошло меньше часа, данные с датчика игнорируются. Только по истечении установленного времени происходит очередной анализ.

*Реакция на освещение*

Освещение играет ключевую роль в жизнедеятельности растения, и "Растение-тамагочи" это учитывает. Когда уровень света достаточный, устройство отображает на дисплее бодрствующее состояние растения. Это сопровождается анимацией моргающих "глаз", создающей иллюзию активности.

Если свет выключается или становится недостаточным, датчик света фиксирует это изменение. Arduino интерпретирует сигнал как команду "перевести растение в режим сна". На дисплее появляется анимация закрытых "глаз", и растение переходит в спящий режим. Оно остаётся в этом состоянии до тех пор, пока свет не вернётся на нужный уровень.

# **Заключение**

Проект "Растение-тамагочи" представляет собой инновационное устройство, которое делает уход за растением более эмоциональным и увлекательным. Система эффективно использует датчики и дисплей для создания иллюзии взаимодействия с растением, что способствует формированию более ответственного и бережного отношения к природе.

Устройство может быть полезным не только для частного использования, но и для образовательных целей. Оно наглядно демонстрирует возможности встроенных систем и электронных компонентов, что делает его подходящим для использования в учебных заведениях.

Кроме того, "Растение-тамагочи" имеет потенциал для дальнейшего развития. Например, можно добавить автоматическую систему полива, контроль температуры или интеграцию с мобильными приложениями для удалённого управления. Эти дополнения сделают устройство ещё более функциональным и удобным.

Таким образом, проект "Растение-тамагочи" является не только оригинальным и увлекательным решением, но и перспективной платформой для изучения и внедрения новых технологий в повседневную жизнь.

# **Схема подключения**

Схема подключения проекта «Растение-тамагочи» на Arduino UNO представлена на рисунке 1.

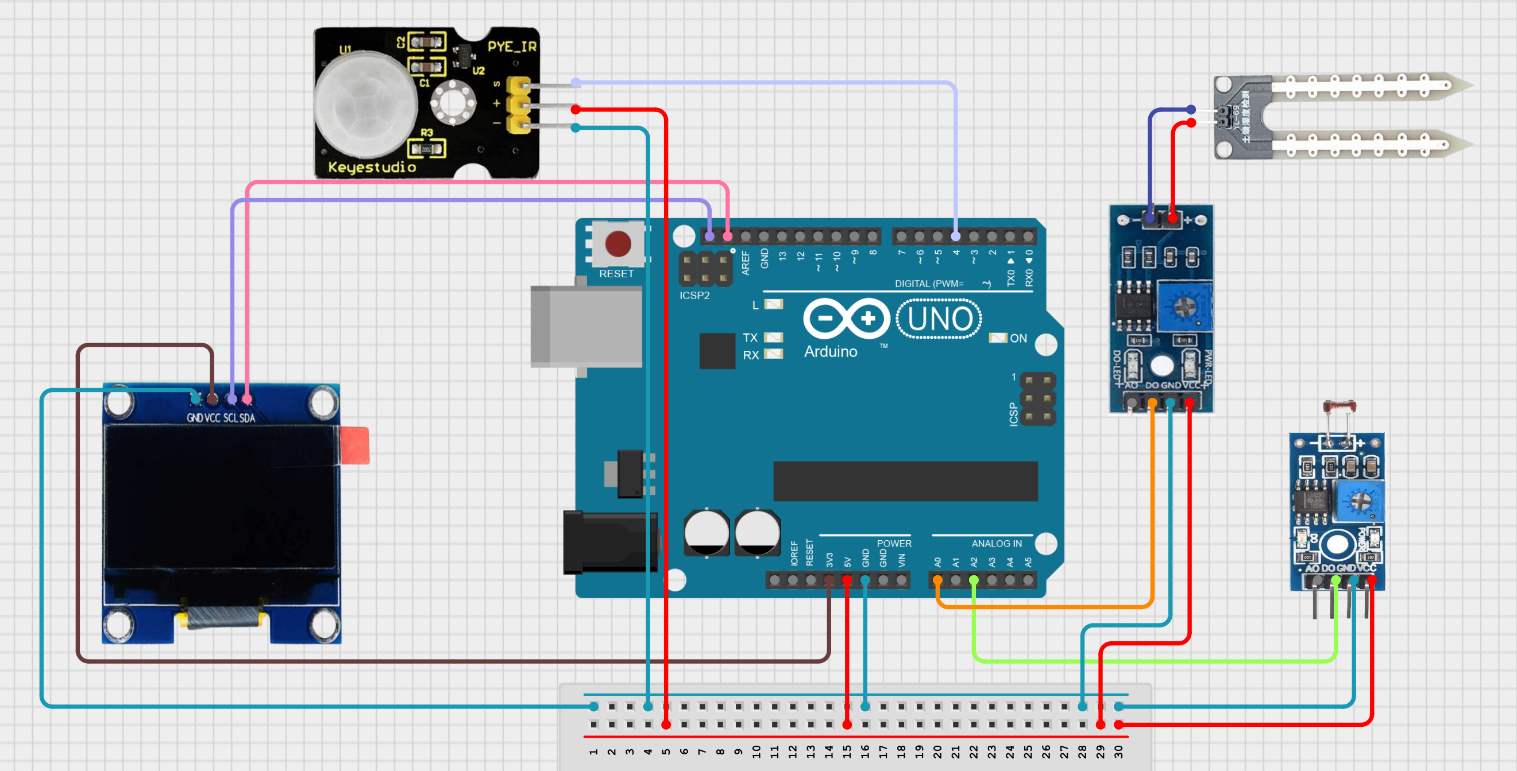
****

Рисунок 1 – Схема подключения проекта «Растение-тамагочи»

# **Листинг кода**

|  |
| --- |
| #include <Wire.h>  #include <Adafruit\_GFX.h>  #include <Adafruit\_SSD1306.h>  #define moisture\_sensor A0  #define SCREEN\_WIDTH 128  #define SCREEN\_HEIGHT 32  #define OLED\_RESET -1  Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);  static const unsigned char PROGMEM eye0[] =  {  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B01111111, B11111111, B11111111, B11111110,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111110,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B01111111, B11111111, B11111111, B11111110,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  };  static const unsigned char PROGMEM eye1[] =  {  B00000111, B11111111, B11111111, B11100000,  B00011111, B11111111, B11111111, B11111000,  B00111111, B11111111, B11111111, B11111100,  B01111111, B11111111, B11111111, B11111110,  B01111111, B11111111, B11111111, B11111110,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B01111111, B11111111, B11111111, B11111110,  B01111111, B11111111, B11111111, B11111110,  B00111111, B11111111, B11111111, B11111100,  B00011111, B11111111, B11111111, B11111000,  B00000111, B11111111, B11111111, B11100000,  };  static const unsigned char PROGMEM eye4[] =  {  B00000111, B11111111, B11111111, B11100000,  B00011111, B11111111, B11111111, B11111000,  B00111111, B11111111, B11111111, B11111100,  B01111111, B11111111, B11111111, B11111110,  B01111111, B11111111, B11111111, B11111110,  B11111111, B11111111, B11111111, B11111111,  B11111111, B11111111, B11111111, B11111111,  B11111000, B00000000, B00000000, B00011111,  B11100000, B00000000, B00000000, B00000111,  B11000000, B00000000, B00000000, B00000011,  B10000000, B00000000, B00000000, B00000001,  B10000000, B00000000, B00000000, B00000001,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  };  static const unsigned char PROGMEM eye5[] =  {  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B01111111, B11111110, B00000000,  B00000001, B11111111, B11111111, B10000000,  B00000011, B11111111, B11111111, B11000000,  B00000111, B11111111, B11111111, B11100000,  B00000111, B11111111, B11111111, B11100000,  B00001111, B11111111, B11111111, B11110000,  B00001111, B10000000, B00000001, B11110000,  B00001110, B00000000, B00000000, B01110000,  B00001100, B00000000, B00000000, B00110000,  B00001000, B00000000, B00000000, B00010000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  B00000000, B00000000, B00000000, B00000000,  };  const int Touch = 4;  const int photoPin = A2;  const int wetPin = A0;  const int moistureThreshold = 700;  int very\_moist\_value = 0;  bool isMessageDisplayed = false;  bool isSleeping = false;  unsigned long lastPressedTime = 0;  unsigned long displayDuration = 500;  void setup() {  pinMode(Touch, INPUT\_PULLUP);  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {  Serial.println(F("SSD1306 allocation failed"));  for (;;);  }  display.clearDisplay();  display.display();  display.drawBitmap(20, 0, eye4, 32, 32, WHITE);  display.drawBitmap(76, 0, eye4, 32, 32, WHITE);  display.display();  delay(1000);  }  void loop() {  static unsigned long lastMoistureCheckTime = 3600000;  unsigned long currentMillis = millis();  int lightLevel = analogRead(photoPin);  int wetLevel = analogRead(wetPin);  if (lightLevel > 200) { //если темно  if (!isSleeping) {  sleepMode();  }  } else { //если светло  isSleeping = false;  //моргание  if (!isMessageDisplayed) {  blinkEyes();  }  //касание  if (digitalRead(Touch) == HIGH && !isMessageDisplayed) {  display.clearDisplay();  isMessageDisplayed = true;  for (int i = 0; i < 2; i++) {  joy();  }  isMessageDisplayed = false;  }  //полив  if (wetLevel < 70 && !isMessageDisplayed && currentMillis - lastMoistureCheckTime >= 3600000) {  lastMoistureCheckTime = currentMillis;  display.clearDisplay();  isMessageDisplayed = true;  joy();  delay(10);  int moisture\_value = analogRead(moisture\_sensor);  display.clearDisplay();  display.setTextSize(2);  display.setTextColor(WHITE);  display.setCursor(3, 10);  display.print("Thank you");  display.display();  delay(10000);  isMessageDisplayed = false;  }  }  }  void closeEyes() {  display.clearDisplay();  display.drawBitmap(20, 0, eye0, 32, 32, WHITE);  display.drawBitmap(76, 0, eye0, 32, 32, WHITE);  display.display();  }  void sleepMode() {  isSleeping = true;  isMessageDisplayed = true;  for (int i = 0; i < 3; i++) {  display.clearDisplay();  display.drawBitmap(20, 0, eye1, 32, 32, WHITE);  display.drawBitmap(76, 0, eye1, 32, 32, WHITE);  display.display();  delay(500);  display.clearDisplay();  display.drawBitmap(20, 0, eye0, 32, 32, WHITE);  display.drawBitmap(76, 0, eye0 ,32 ,32 ,WHITE);  display.display();  delay(700);  }  closeEyes();  while (analogRead(photoPin) > 200) {  animateZs();  }  for (int i = 0; i < 2; i++) {  display.clearDisplay();  display.drawBitmap(20, 0, eye0, 32, 32, WHITE);  display.drawBitmap(76, 0, eye0 ,32 ,32 ,WHITE);  display.display();  delay(700);  display.clearDisplay();  display.drawBitmap(20, 0, eye1, 32, 32, WHITE);  display.drawBitmap(76, 0, eye1, 32, 32, WHITE);  display.display();  delay(200);  }  isMessageDisplayed = false;  }  void blinkEyes() {  display.clearDisplay();  display.drawBitmap(20, 0, eye1, 32, 32, WHITE); // Левый глаз  display.drawBitmap(76, 0, eye1, 32, 32, WHITE); // Правый глаз  display.display();  delay(800);  display.clearDisplay();  display.drawBitmap(20, 0, eye0, 32, 32, WHITE); // Левый глаз  display.drawBitmap(76, 0, eye0 ,32 ,32 ,WHITE); // Правый глаз  display.display();  delay(200);  }  void joy() {  display.clearDisplay();  display.drawBitmap(20, 0, eye4, 32, 32, WHITE); // Левый глаз  display.drawBitmap(76, 0, eye4, 32, 32, WHITE); // Правый глаз  display.display();  delay(500);  display.clearDisplay();  display.drawBitmap(20, 0, eye5, 32, 32, WHITE); // Левый глаз  display.drawBitmap(76, 0, eye5, 32, 32, WHITE); // Правый глаз  display.display();  delay(500);  }  void animateZs() {  for (int i = 0; i < 3; i++) {  display.clearDisplay();    int eyeOffset = (i % 2 == 0) ? -1 : 1;  display.drawBitmap(20, 0 + eyeOffset, eye0, 32, 32, WHITE);  display.drawBitmap(76, 0 + eyeOffset, eye0, 32, 32, WHITE);    display.setTextSize(i + 0.05);  display.setTextColor(WHITE);  display.setCursor(100 + i \* 2, 5);  display.print("Z");    display.display();  delay(300);  }  } |