

«Учебный день в ИТ-полигоне»

Программирование устройств на Arduino

Название кейса: Система умного освещения.

Формулировка задания: Разработать систему умного освещения, реагирующую на движение. Требуется собрать и запрограммировать устройство, которое включают светодиодную лампочку, если датчик фиксирует движение при низком уровне освещения и выключает свет, если движения нет. Система должна выводить показания уровня освещенности и оповещение о том, что свет включился на LCD дисплей. Также должна быть реализована возможность управления светом с кнопки.

Необходимое оборудование, расходные материалы, ПО:

- 1 плата Arduino Uno
- 1 беспаячная макетная плата
- 1 светодиод
- 1 фоторезистор
- 1 пирозлектрический инфракрасный (PIR) датчик движения
- 1 ползунковый переключатель
- 1 LCD дисплей
- 3 резистора номиналом 1 кОм
- набор проводов «папа-папа»
- среда разработки Arduino IDE
- компьютер

Рабочие материалы для учащихся (карточки, инструкции, тестовые материалы и т.п.)

Теоретическая часть

Термин “ умный свет ” относится к сфере управления системами контроля освещения. Речь идет о настенных, потолочных и точечных светильниках, садовых фонарях, светодиодных подсветках, которые могут управляться как автоматически, так и дистанционно. В качестве датчиков системы чаще всего являются датчики освещения, различны типы датчиков присутствия, GPS - приемники. По всему земному шару проблемы энергосбережения считаются одними из наиболее важных проблем, и эффективное, или смарт-освещение, вносит значительный вклад в деле сохранения энергии.

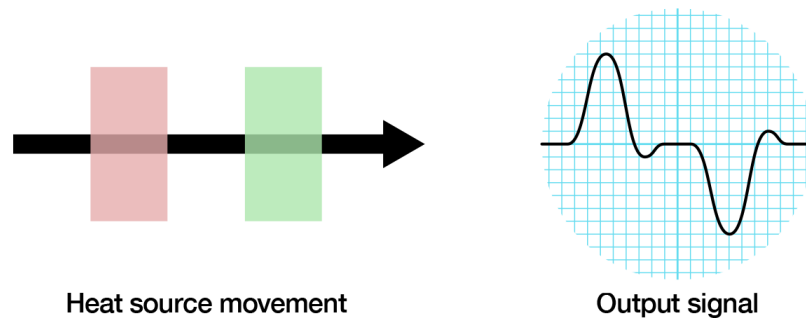
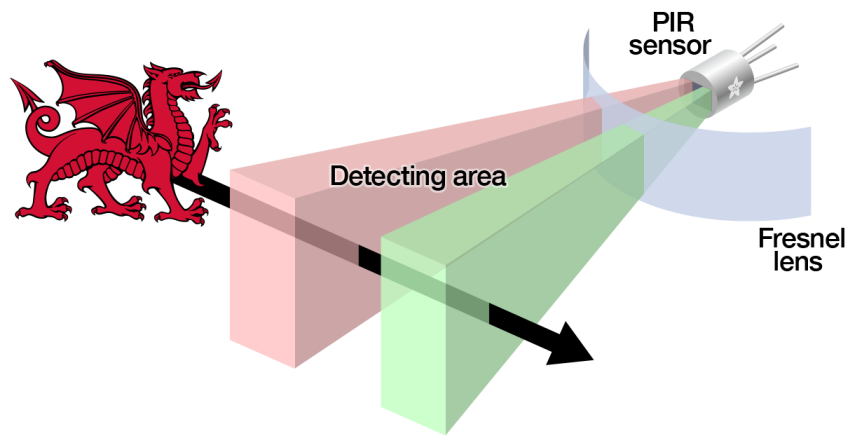
Данная система поможет эффективно использовать электроэнергию. При приближении объекта к системе, загорается свет, каждые 6 секунд она проверяет на наличие объекта (то есть человека) в поле видимости устройства. Тем самым свет будет включаться только при необходимости, а

также не нужно будет заботиться о не выключенном свете, так как устройство само его выключит. Это система будет актуальна для складских помещений или помещений с похожим применением, где не нужен постоянный свет (т.е. только при необходимости освещения).

Основой программно-аппаратного комплекса является плата Arduino Uno и пироэлектрический инфракрасный (PIR) датчик движения. PIR датчики движения по сути состоят из пироэлектрического чувствительного элемента (цилиндрическая деталь с прямоугольным кристаллом в центре), который улавливает уровень инфракрасного излучения. Все вокруг излучает небольшой уровень радиации. Чем больше температура, тем выше уровень излучения. Датчик фактически разделен на две части. Это обусловлено тем, что нам важен не уровень излучения, а непосредственно наличие движение в пределах его зоны чувствительности. Две части датчика установлены таким образом, что если одна половина улавливает больший уровень излучения, чем другая, выходной сигнал будет генерировать значение high или low.

Пироэлектрический датчик движения состоит из двух основных частей. Каждая из частей включает в себя специальный материал, чувствительный к инфракрасному излучению. В данном случае линзы особо не влияют на работу датчика, так что мы видим два участка чувствительности всего модуля. Когда датчик находится в состоянии покоя, оба сенсора определяют одинаковое количество излучения. Например, это может быть излучение помещения или окружающей среды на улице. Когда теплокровный объект (человек или животное), проходит мимо, он пересекает зону чувствительности первого сенсора, в результате чего на модуле ПИР датчика генерируются два различных значения излучения. Когда человек покидает зону чувствительности первого сенсора, значения выравниваются. Именно изменения в показаниях двух датчиков регистрируются и генерируют импульсы HIGH или LOW на выходе.



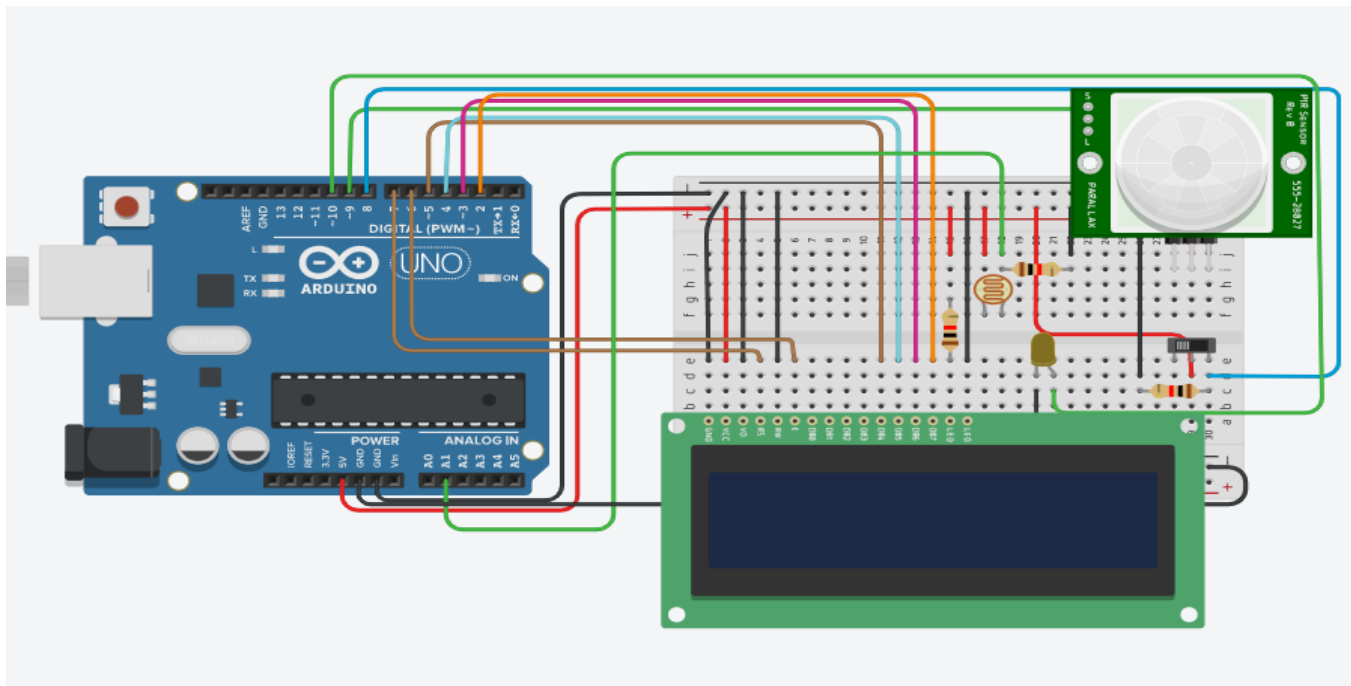


Сборка схемы

Для начала соберем схему устройства, используя безопасную макетную плату и набор электронных компонентов:

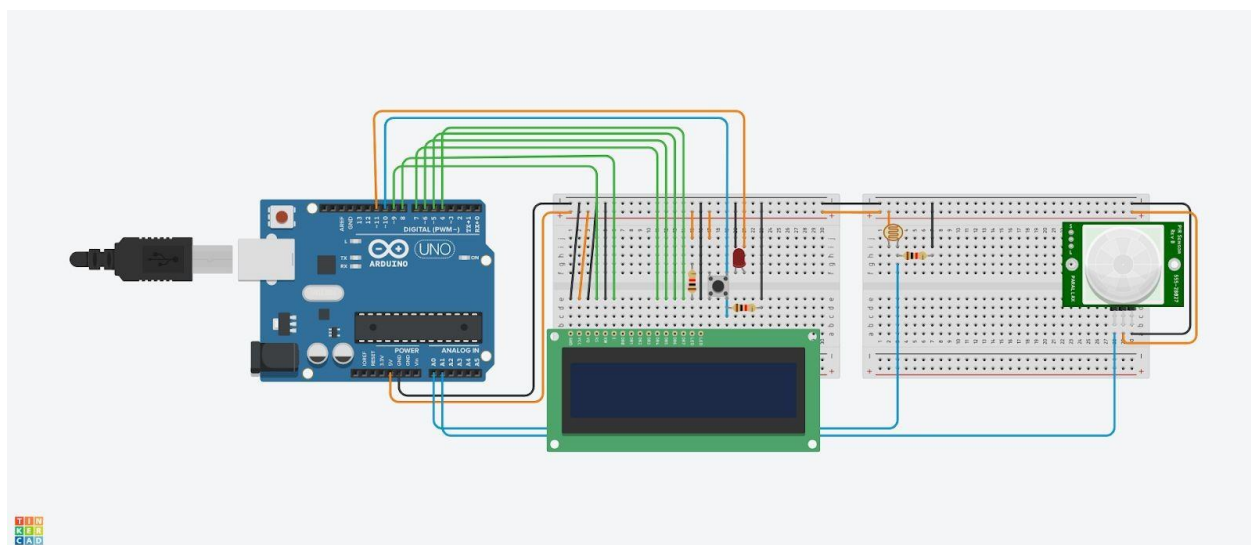
- 1 плата Arduino Uno
- 1 безопасная макетная плата
- 1 светодиод
- 1 фоторезистор
- 1 пирозлектрический инфракрасный (PIR) датчик движения
- 1 ползунковый переключатель
- 1 LCD дисплей
- 3 резистора номиналом 1 кОм
- набор проводов «папа-папа»

Возьмите подготовленные компоненты и соберите схему на макетной плате, изображенную на рисунке ниже.



Вариант 1. Схема подключения устройства на макетной плате

Для удобства проектирования можно выбрать реализацию с двумя макетными платами. На первой плате находится светодиод, кнопка и дисплей. На второй: датчик освещенности и датчик движения.



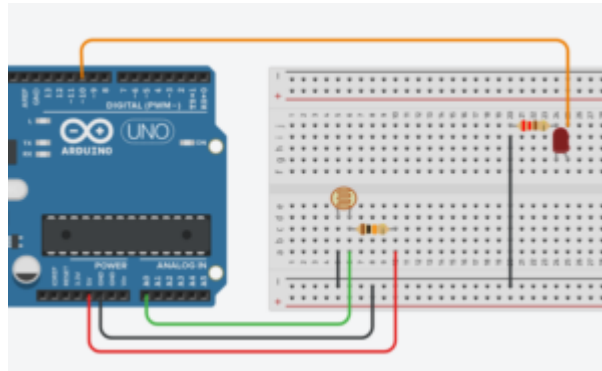
Вариант 2. Схема подключения устройства на макетной плате

Для программирования устройства рекомендуется воспользоваться данной информацией

Пример скетча датчика освещенности на фоторезисторе

Мы подключили схему с фоторезистором к ардуино, убедились, что все сделали правильно. Теперь осталось запрограммировать контроллер.

Написать скетч для датчика освещенности довольно просто. Нам нужно только снять текущее значение напряжения с того аналогового пина, к которому подключен датчик. Делается это с помощью известной нам всем функции `analogRead()`. Затем мы можем выполнять какие-то действия, в зависимости от уровня освещенности.



Давайте напишем скетч для датчика освещенности, включающего или выключающего светодиод, подключенный по следующей схеме.

Алгоритм работы таков:

- Определяем уровень сигнала с аналогового пина.
- Сравниваем уровень с пороговым значением. Максимально значение будет соответствовать темноте, минимальное – максимальной освещенности. Пороговое значение выберем равное 300.
- Если уровень меньше порогового – темно, нужно включать светодиод.
- Иначе – выключаем светодиод.

```
#define PIN_LED 13

#define PIN_PHOTO_SENSOR A0

void setup() {

  Serial.begin(9600);

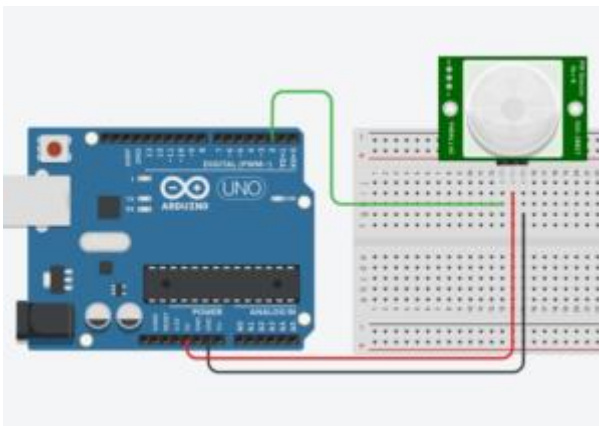
  pinMode(PIN_LED, OUTPUT);

}
```

```
void loop() {  
  
  int val = analogRead(PIN_PHOTO_SENSOR);  
  
  Serial.println(val);  
  
  if (val < 300) {  
  
    digitalWrite(PIN_LED, LOW);  
  
  } else {  
  
    digitalWrite(PIN_LED, HIGH);  
  
  }  
  
}
```

Прикрывая фоторезистор (руками или светонепроницаемым предметом), можем наблюдать включение и выключение светодиода. Изменяя в коде пороговый параметр, можем заставлять включать/выключать лампочку при разном уровне освещения.

Схема подключения датчика движения к Ардуино



Подключение PIR-датчика к Ардуино

выполнить не сложно. Чаще всего модули с сенсорами движения оснащены тремя коннекторами на задней части. Распиновка каждого устройства зависит от производителя, но чаще всего возле выходов есть соответствующие надписи. Поэтому, прежде чем выполнить подключение датчика к Arduino необходимо ознакомиться с обозначениями. Один выход идет к земле (GND), второй – обеспечивает выдачу необходимого сигнала с сенсоров (+5V), а третий является цифровым выходом, с которого снимаются данные.

Подключение PIR-сенсора:

- «Земля» – на любой из коннекторов GND Arduino;
- Цифровой выход – на любой цифровой вход или выход Arduino;
- Питание – на +5В на Arduino.

Пример программы

Скетч представляет собой программный код, который помогает проверить работоспособность датчика движения после его включения. В самом простом его примере есть множество недостатков:

- Вероятность ложных срабатываний, за счет того, что для самоинициализации датчика требуется одна минута;
- Отсутствие выходных устройств исполнительного типа – реле, сирены, светоиндикации;
- Короткий временной интервал сигнала на выходе сенсора, который необходимо на программном уровне задержать, в случае появления движения.

Указанные недостатки устраняются при расширении функционала датчика.

Скетч самого простого типа, который может быть использован в качестве примера работы с датчиком движения на Arduino, выглядит таким образом:

```
#define PIN_PIR 2

#define PIN_LED 13

void setup() {

  Serial.begin(9600);

  pinMode(PIN_PIR, INPUT);

  pinMode(PIN_LED, OUTPUT);
```

```
}  
  
void loop() {  
  
    int pirVal = digitalRead(PIN_PIR);  
  
    Serial.println(digitalRead(PIN_PIR));  
  
    //Если обнаружили движение  
  
    if (pirVal)  
  
    {  
  
        digitalWrite(PIN_LED, HIGH);  
  
        Serial.println("Motion detected");  
  
        delay(2000);  
  
    }  
  
    else  
  
    {  
  
        //Serial.print("No motion");  
  
        digitalWrite(PIN_LED, LOW);  
  
    }  
  
}
```

Авторское решение кейса (программы, фотографии сборок, видео-инструкции, 3D-модели и т.п.)

Программирование. Реализация функционала устройства.

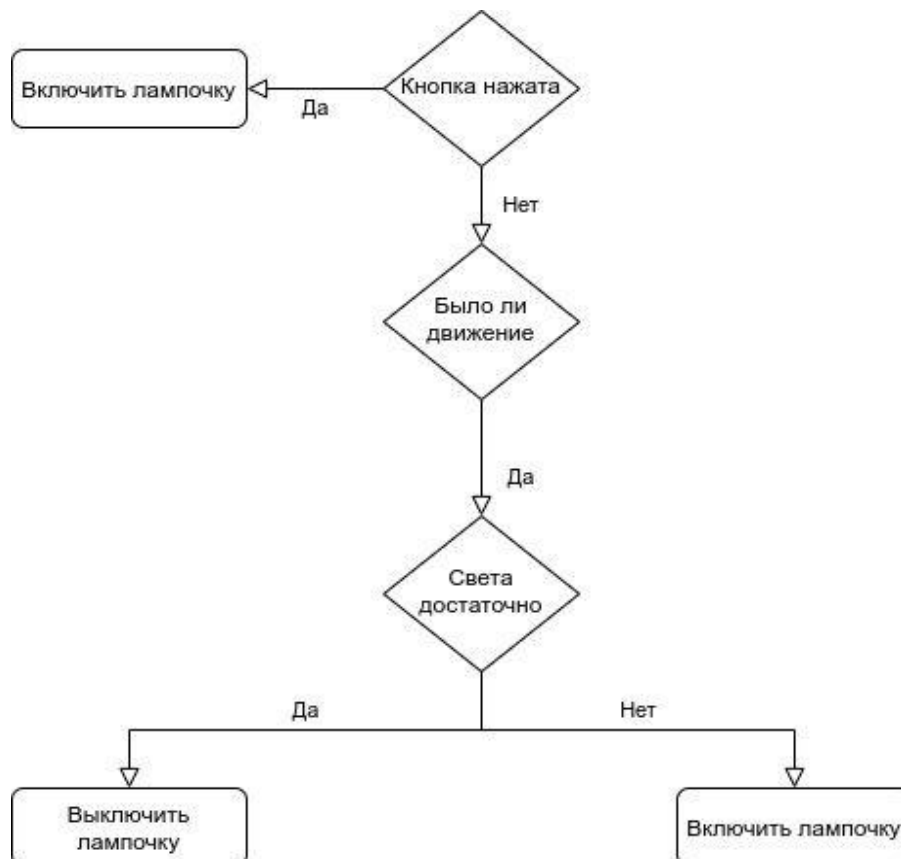
После сборки системы (вариант 1) запрограммируем ее на языке C с учётом возможных следующих модификаций. Тщательно проверив все аспекты модели, можно начать работу над практической реализацией:


```

1  #include <LiquidCrystal.h>
2
3  int rs=7, en=6, d4=5, d5=4, d6=3, d7=2;
4  LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
5  int LED=10;
6  int photo=A1;
7  int movement=9;
8  int LightOn=8;
9  bool Switch=false;
10
11 void setup()
12 {
13   lcd.begin(16,2);
14   pinMode (LED, OUTPUT) ;
15   pinMode (movement, INPUT);
16   pinMode (LightOn, INPUT);
17   Serial.begin(9600);
18 }
19
20 void loop()
21 {
22
23   lcd.clear();
24   int sun=analogRead(photo);
25   lcd.print(sun);
26   lcd.print(" ");
27   lcd.setCursor(0,1);
28   Switch = digitalRead(LightOn);
29
30   if (Switch)
31   {
32     lcd.print("Lights On");
33     analogWrite (LED, 1020);
34   }
35   else
36   {
37     if (digitalRead (movement) ==1)
38     {
39       if (sun<400)
40       {
41         lcd.print ("lights On");
42         analogWrite (LED, 1020);
43       }
44     }
45     else
46     {
47       lcd.print ("Enough sun");
48       analogWrite (LED, 0);
49     }
50   }
51 }
52 else
53 {
54   lcd.print("lights off");
55   analogWrite (LED, 0);
56 }
57 }
58 }
59 }
60 delay(6000);
61 }
--

```

Листинг программы для микроконтроллера

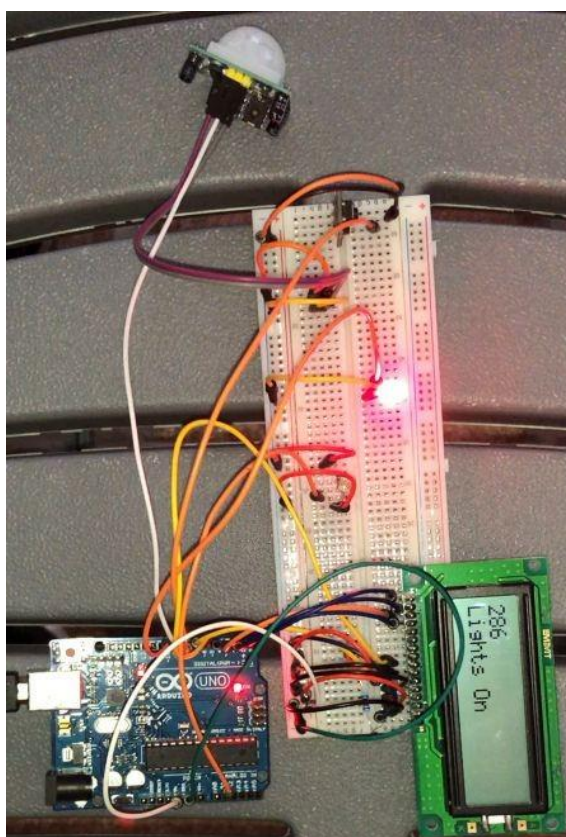
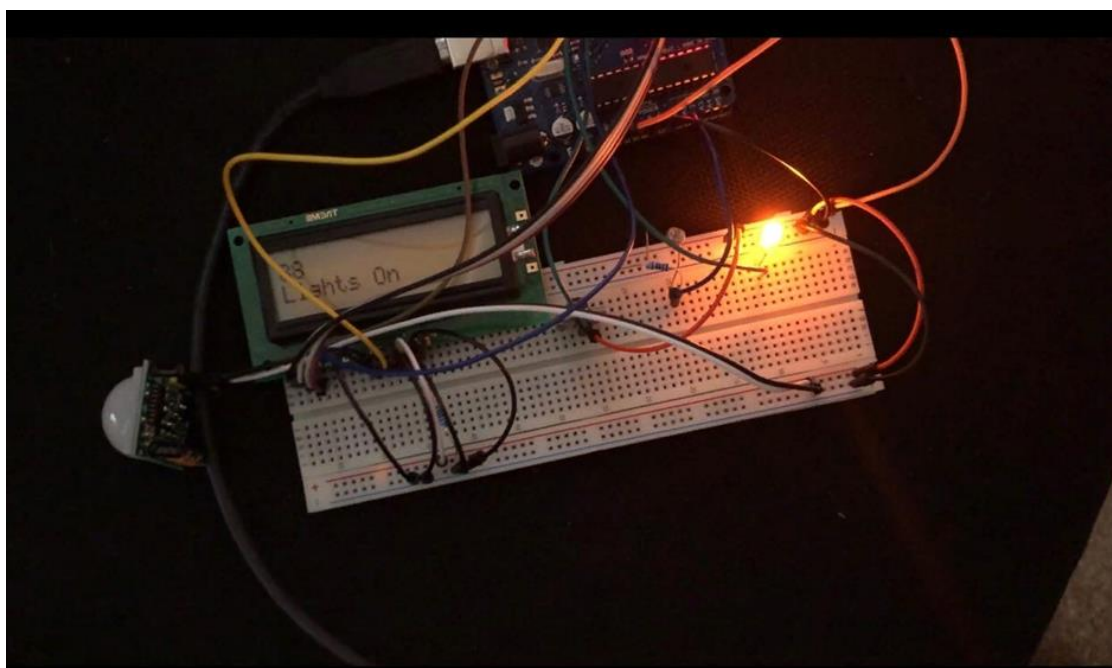


Алгоритм работы программного кода

Сборка прототипа устройства

После проверки кода и модели в программе Tinkercad нужно собрать прототип. При сборке прототипа требуется учесть возможные ошибки виртуальной схемы. Для реализации прототипа была собрана схема, а также загружен программный код на микроконтроллер Arduino Uno. Видео работы устройства по ссылке:

https://drive.google.com/file/d/1WzfPhO2tTewRoYLGcCDNFv_aaIyd-s4N/view?usp=sharing



**Чек-лист проверки выполнения задания с указанием критериев и
уровневой оценки (чек-лист должен учитывать предметные знания,
предпрофессиональные умения, мягкие навыки):**

Задание	Критерии оценивания	Max баллы	Результат
1.	Схема собрана в среде Tinkercad	5	
2.	Умение определять элементы схемы, правильность подключения выходов/входов питания, сигнала.	10	
3.	Светодиод горит при включенном тумблере	5	
4.	Система выводит показания уровня освещенности на LCD дисплей	5	
5.	Система выводит оповещения на LCD дисплей	5	
6.	Пирозэлектрический датчик реагирует на движение	5	
7.	Фоторезистор работает – светодиод включается и отключается согласно правилу	5	
8.	Собран прототип устройства (в реальности) на платформе Arduino Uno	10	
9.	Умение разделять обязанности между членами команды	10	
10.	Умение работать с информацией	10	
11.	Внесение в проект собственных решений	10	
12.	Программирование устройства выполнено самостоятельно	20	
13.	Программирование устройства выполнено с помощью учителя	5	
14.	Программирование устройства выполнено с помощью ребят-консультантов из другой группы	10	

Общая оценка определяется суммой всех оценок из чек-листа.

Отличный уровень: 75-100 баллов

Хороший уровень: 55- 74 баллов

Удовлетворительный уровень: 5-54 баллов.