

ПРИМЕРЫ РЕАЛИЗАЦИИ ПОДСИСТЕМ «УМНОГО ДОМА» НА ПЛАТЕ ARDUINO

Е. Р. Кирколуп, А. А. Кудрявцева

Алтайский государственный технический университет
им. И.И. Ползунова, г. Барнаул

В статье рассмотрены примеры подсистем «умного дома», реализованных на плате Arduino. Приведено описание подсистем контроля температуры и влажности, обнаружения протечек воды, пожарной сигнализации. Представлен вариант использования данных подсистем в загородном одноэтажном жилом здании.

Ключевые слова: умный дом, плата Arduino, подсистема контроля температуры и влажности, подсистема обнаружения протечек воды, пожарная сигнализация.

«Умный дом» – это набор управляемых автоматизированных подсистем, объединенных в одну общую систему, которая обеспечивает согласованное функционирование инженерных сетей здания [1, с. 170]. К системе «умный дом» относят следующие подсистемы управления зданием [2, с. 6]:

- климат контроль (автоматизированное отопление, вентиляция, кондиционирование);
- аварийный контроль (контроль утечек газа, протечек воды);
- системы безопасности (видеонаблюдение, охранно-пожарная сигнализация, контроль доступа в помещения);
- автоматизированное освещение помещений и здания в целом (включая датчики движения, звука, освещенности и др.);
- автоматизированное электропитание здания (включая альтернативные источники энергии, источники бесперебойного питания, различные тепловые генераторы);
- автоматизированное управление различными электроприводами (автоматические ворота, шлагбаумы и т.п.);
- управление мультимедиа-, аудио- и видеоаппаратурой;
- системы удаленного управления и мониторинга здания (телеметрии, IP-мониторинга, GSM-мониторинга).

Комплексная установка всех подсистем «умного дома» является слишком затратной, поэтому такие системы в России пользуются меньшей популярностью. Однако если рассматривать подсистемы «умного дома» по отдельности, то в каждом конкретном случае можно выбрать только на самом деле нужные компоненты, которые будут непременно использоваться. Модульность в построении систем «умного дома» в разы снижает ее

стоимость, поэтому данный подход активно практикуется и российскими и зарубежными производителями.

При проектировании подсистем «умного дома» важно выбрать технологию такого рода, с помощью которой можно было бы строить систему, используя модульный принцип, и которая позволяла бы создавать беспроводные, серверные и прочие системы. Самым простым вариантом для этого может служить программируемый микроконтроллер и набор различных датчиков. Из имеющегося разнообразия предлагаемых решений существуют такие, которые позволяют проектировать различные устройства для подсистем «умного дома», при этом легко программируемы и доступны по стоимости. К таким решениям можно отнести микроконтроллеры семейств ARM, AVR, MCS 51 на бюджетных платах Arduino, Iskra Neo, Raspberry Pi и других. В данной работе приведены примеры построения подсистем «умного дома» на плате Arduino для их использования, например, в загородном одноэтажном жилом здании.

В качестве первого примера рассмотрим подсистему контроля температуры и влажности, выполненной на плате Arduino Uno с использованием датчика температуры и влажности DHT11. Датчик DHT11 имеет три контакта. Левый контакт соединяется с аналоговым входом платы (например, с контактом A0), на центральный подается напряжение с платы (контакт 5V), а правый заземляется (контакт GND). Температура и влажность передаются датчиком на плату через левый контакт по специальному протоколу. Принцип передачи данных приблизительно такой: сначала микроконтроллер подает сигнал о готовности считывать показания (на контакт подается 0, а через некоторое время – 1).

Датчик аналогичными действиями подтверждает готовность к передаче данных. Затем датчик передает последовательность из 5 байт, которая содержит в первых двух байтах значение температуры, в следующих двух байтах – значение влажности, а в пятом байте передается контрольная сумма для проверки отсутствия ошибок считывания. Такая работа датчика менее энергозатратна, т.е. пока нет запроса от микроконтроллера, датчик находится в режиме ожидания и практически не потребляет ток.

Для программирования датчика DHT11 можно использовать, например, среду программирования Arduino IDE. Тогда программирование датчика сведется к подключению библиотеки DHT.h, специально разработанной для данной серии датчиков, и заданию параметров ввода и вывода информации.

Приведем пример программного кода, позволяющего считывать показания с датчика и передавать их на монитор порта Arduino IDE.

```
// Контроль температуры и влажности
#include <DHT.h>
#define dhtpin A0
DHT dht(dhtpin, DHT11);
void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  delay(5000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  String ts = String(t);
  String hs = String(h);
  if (isnan(t) || isnan(h)) {
    Serial.println("Ошибка чтения данных с датчика DHT");
  } else {
    String datastring = "Температура =" + ts.substring(0,2) + "\xB0" + "C, " + "Влажность =" + hs.substring(0,2) + "%.";
    Serial.println(datastring);
  }
}
```

В качестве другого примера приведем описание подсистемы обнаружения протечек воды, построенной на датчике уровня жидкости (Water Sensor). Напряжение питания датчика равно 5 В. Выходное напряжение датчика зависит от глубины его погружения и от рода жидкости.

Для воды показания датчика варьируются от 0 (сухой) до 1023 (полностью погружен).

При подключении датчика уровня жидкости к плате Arduino левый контакт (S) соединяется с аналоговым входом платы (A1), центральный (+) – с контактом 5V, а правый (-) – с заземлением (GND). В качестве сигнального оповещения для данной подсистемы следует предусмотреть подключение светодиода или зуммера KY-012.

Для подсистемы обнаружения протечек воды со светодиодным индикатором можно использовать следующий программный код:

```
// Обнаружение протечек воды
#define IWater A1
#define LED 13
void setup() {
  Serial.begin(9600);
  pinMode(IWater, INPUT);
  pinMode(LED, OUTPUT);
}
void loop() {
  delay(5000);
  String datastring = "Water level: "+ String(analogRead(IWater));
  Serial.println(datastring + '.');
  int x = analogRead(IWater);
  if (x>100) {
    digitalWrite(LED, 200);
  } else {
    digitalWrite(LED, LOW);
  }
}
```

Рассмотрим еще одну подсистему «умного дома», которая может служить своеобразной системой пожарной сигнализации, собранной на плате Arduino с использованием датчика пламени YS-17 (KY-026) и зуммера KY-012.

Система пожарной сигнализации может быть построена по следующей схеме (рисунок 1) [3]. Датчик пламени подключается к аналоговому входу платы и на него подается напряжение с этой платы. Зуммер служит в этой схеме источником звукового сигнала и подключается к одному из цифровых выходов. Кроме этого в данной схеме предусмотрена система цветовой индикации из трех светодиодов (LED): красного, зеленого и синего, которые также подключаются к цифровым выходам платы. Чувствительность датчика пламени и яркость светодиодов регулируется с помощью резисторов номиналом 20 кОм и 220 Ом.

Работает система в выключенном и включенном состояниях. В выключенном состоянии на каждое появление источника огня вблизи датчика загорается синий индикатор.

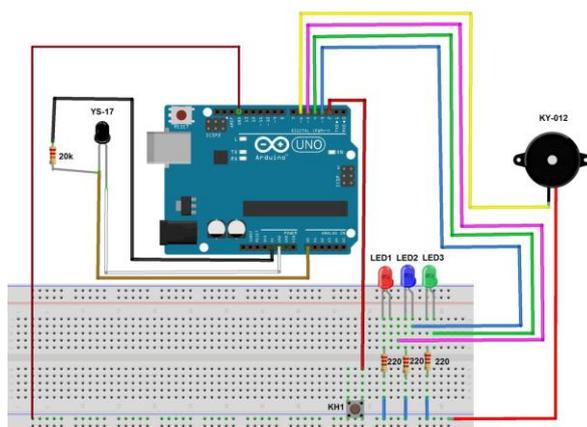
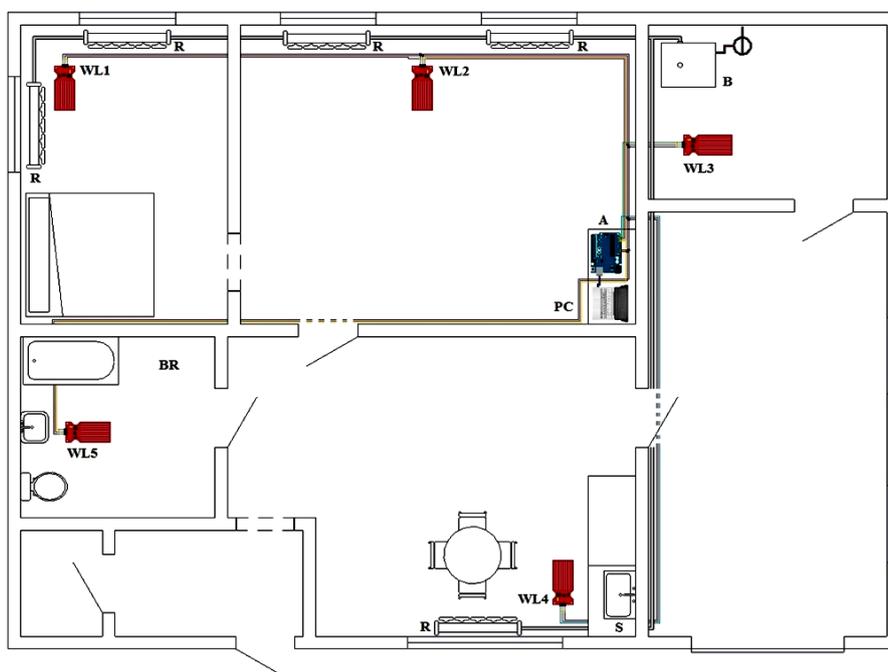


Рисунок 1 – Схема пожарной сигнализации на Arduino



Условные обозначения: А – плата Arduino UNO, В – газовый котел, BR – ванная комната, PC – персональный компьютер, R – радиатор отопления, S – кухонная мойка, WL – датчик уровня жидкости

Рисунок 2 – Схема размещения подсистемы обнаружения протечек воды в загородном одноэтажном жилом здании

Когда система включена, то постоянно горит зеленый индикатор. В момент появления пламени начинает мигать красный индикатор, мигание которого сопровождается звуковым сигналом. Для включения и выключения сигнализации предусмотрена кнопка (KN1).

Приведем пример программного кода, позволяющего работать системе пожарной сигнализации в описанном режиме.

//Пожарная сигнализация

```
bool _trgt1 = 0;
bool _trgt1l = 0;
bool _trgs1 = 0;
```

```
bool _gen1l = 0;
bool _gen1O = 0;
unsigned long _gen1P = 0UL;
bool _bounseInputD2S = 0;
bool _bounseInputD2O = 0;
unsigned long _bounseInputD2P = 0UL;
void setup()
{
  pinMode(2, INPUT);
  digitalWrite(2, HIGH);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(3, OUTPUT);
```

```

    _bouncelInputD2O = digitalRead(2);
}
void loop()
{
    bool _bouncelInputTmpD2 = (digitalRead
(2));
    if (_bouncelInputD2S)
    {
        if (millis() >= (_bouncelInputD2P + 40))
            {_bouncelInputD2O=
_bouncelInputTmpD2; _bouncelInputD2S=0;}
        }
        else
        {
            if (_bouncelInputTmpD2 !=
_bouncelInputD2O )
                {_bouncelInputD2S=1;
_bouncelInputD2P = millis();}
            }
            if((( (analogRead (0))) < (700)) && (_trgt1
)) _trgs1 = 1;
            if(!(_trgt1)) _trgs1 = 0;
            if (_trgs1) { if (! _gen1l) { _gen1l = 1;
_gen1O = 1; _gen1P = millis(); } } else { _gen1l
= 0; _gen1O = 0;}
            if (_gen1l) { if ( _isTimer ( _gen1P , 500 ))
{ _gen1P = millis(); _gen1O = ! _gen1O;}}
            bool _tmp1 = !(_bouncelInputD2O);
            if (_tmp1) { if (! _trgt1l) _trgt1 = ! _trgt1; }
            _trgt1l = _tmp1;
            digitalWrite(3, ( (analogRead (0))) < (700));
            digitalWrite(5, _gen1O);
            digitalWrite(6, _gen1O);
            digitalWrite(4, _trgt1);
            }
            bool _isTimer(unsigned long startTime, un-
signed long period )
            {
                unsigned long currentTime;
                currentTime = millis();
                if (currentTime>= startTime) {return
(currentTime>=(startTime + period));} else {re-
turn (currentTime >=(4294967295-
startTime+period));}
            }
}

```

В случае использования какой-либо из описанных подсистем в реальных условиях, например, в загородном одноэтажном жилом здании, необходимо учитывать число одновременно подключаемых датчиков к плате с микроконтроллером, каким образом будет осуществляться питание платы и мониторинг показаний датчиков. Так при проектировании

подсистемы контроля протечек воды для одноэтажного жилого здания достаточным будет использование 5 датчиков уровня жидкости (рисунок 2), подключенных к плате Arduino Uno. При этом плата подключается к компьютеру, с помощью которого осуществляется мониторинг и питание системы, а датчики располагают в местах потенциальных угроз протечек воды: в ванной комнате, в котельной, на кухне, в жилых комнатах рядом с радиаторами отопления. Если необходимо подключить еще одну или несколько подсистем, или дополнительный набор датчиков, то следует использовать другую плату с микроконтроллером, например, Arduino Mega либо специальные платы расширения.

В заключении отметим, что с помощью набора датчиков и платы Arduino можно спроектировать большое количество систем «умного дома», при этом системы могут быть как простыми, так и сложными, включая серверные, беспроводные, комплексные и прочие системы. Примеры, описанные в работе, представляют собой простые решения для создания и использования подсистем «умного дома» в загородном одноэтажном жилом здании.

СПИСОК ЛИТЕРАТУРЫ

1. Тускаева, З. Р. Система «умный дом» сегодня и завтра / З. Р. Тускаева, Р. А. Дзобоев // Научные исследования и разработки молодых ученых. – 2016. – № 9-1. – С. 170-173.
2. Благодаров, Д. А. Интеллектуальное управление умным домом / Д. А. Благодаров, Е. С. Багаев, Ю. М. Сафонов, А. А. Копесбаева // Потенциал современной науки. – 2016. – № 9 (26). – С. 5-8.
3. Обзор датчика огня YS-17 – Пожарная сигнализация на Arduino [Электронный ресурс] // Arduinoprom.ru – блог Чилингаряна Грачика. – 2015. URL: <http://arduinoprom.ru/obzory-modulej/180-obzor-datchika-ognja-ys-17-pozharnaja-signalizacija-na-arduino.html> (дата обращения: 20.03.2017).

Кирколуп Е.Р. – к.т.н., доцент кафедры «Строительных конструкций» ФГБОУ ВО АлтГТУ им. И.И. Ползунова, E-mail: kirkolup@mail.ru.

Кудрявцева А.А. – магистрант ФГБОУ ВО АлтГТУ им. И.И. Ползунова, E-mail: kudryavceva.alena.a@yandex.ru.