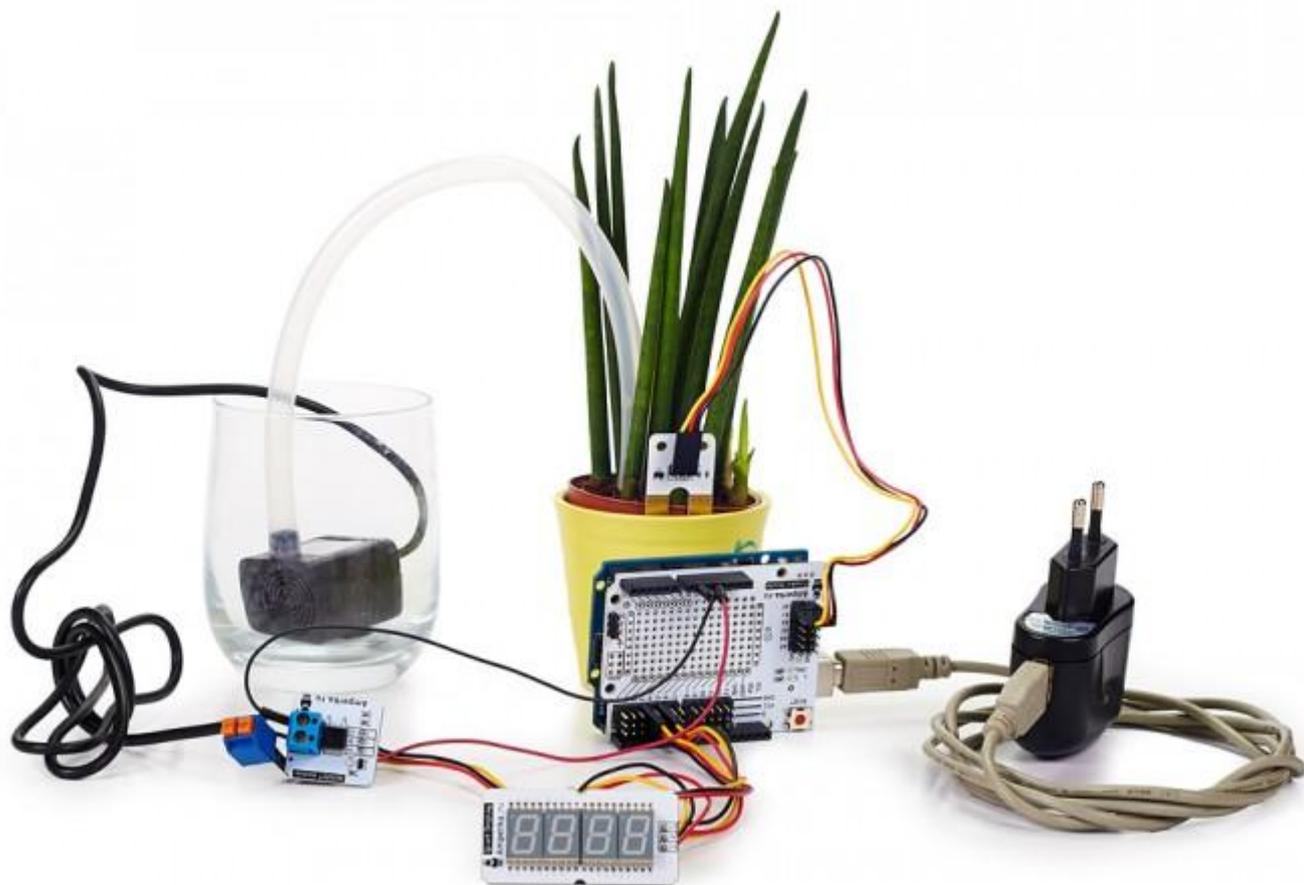


Автополивщик растений на Arduino

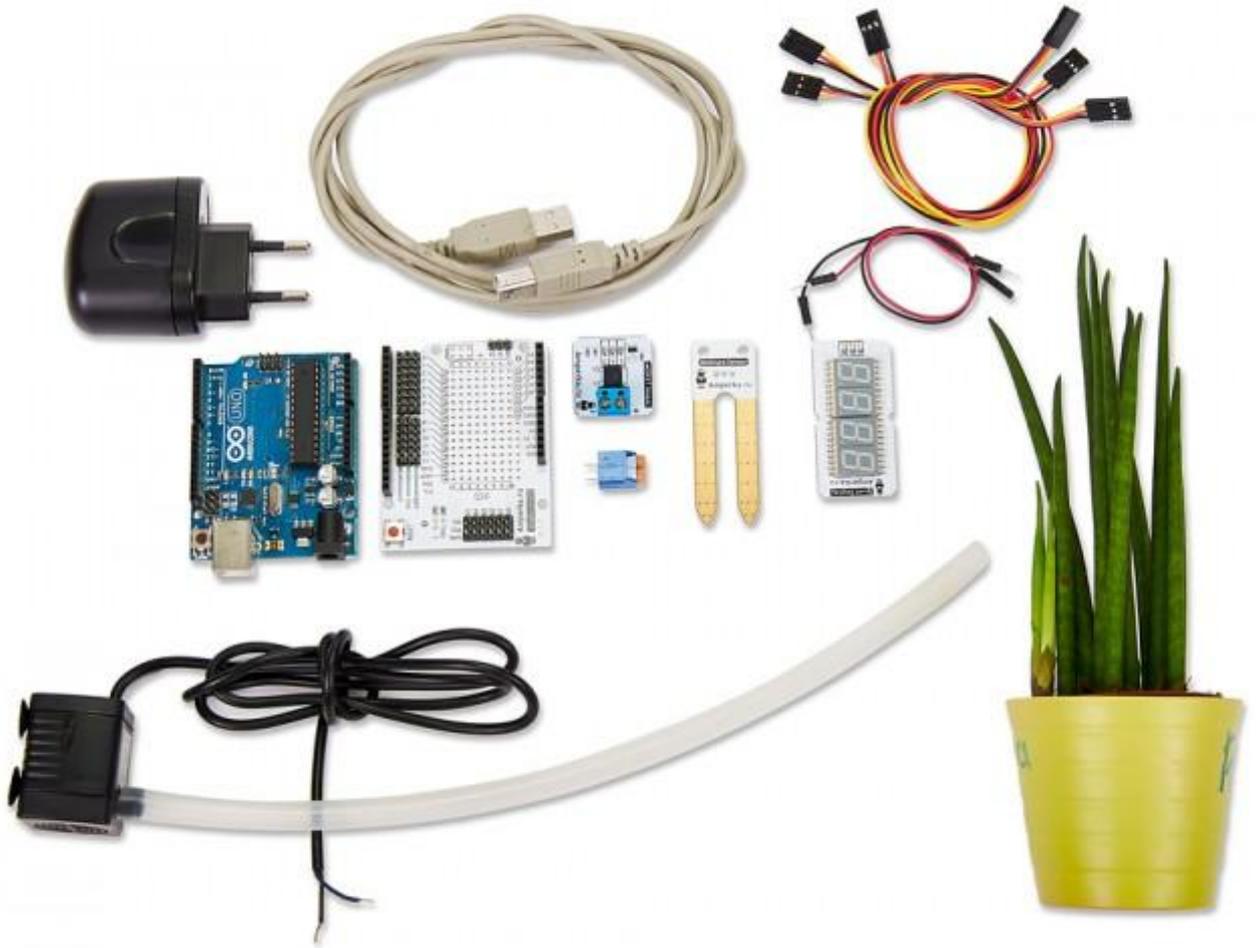


- Платформы: Arduino Uno
- Языки программирования: [Wiring \(C++\)](#)
- Проект на [GitHub](http://github.com/amperka-projects/irrigator): <http://github.com/amperka-projects/irrigator>
- Тэги: автополив, ирригация, гидропоника

Что это такое?

В этой статье мы расскажем о том, как собрать устройство для автоматического полива с контролем влажности почвы — *ирригатор*. Необходимость полива будем определять по показаниям датчика влажности почвы. Одновременно можно будет поливать несколько растений.

Что для этого необходимо?



Для изготовления ирригатора нам понадобятся:

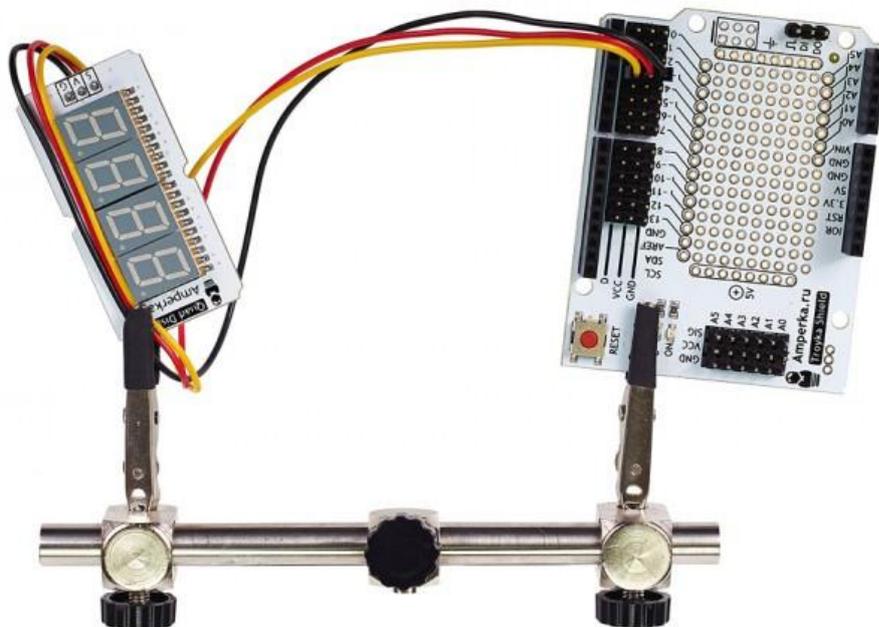
1. Arduino Uno
2. Водяная помпа
3. Сенсор влажности почвы со шлейфом
4. Силовой ключ (Тройка-модуль) со шлейфом
5. Клеммник нажимной
6. Провод «папа-папа» × 1 шт
7. Провод «мама-папа» × 1 шт
8. Блок питания с USB разъёмом
9. USB кабель
10. Растение в горшке с сухой землёй

Так же удобно для индикации использовать:

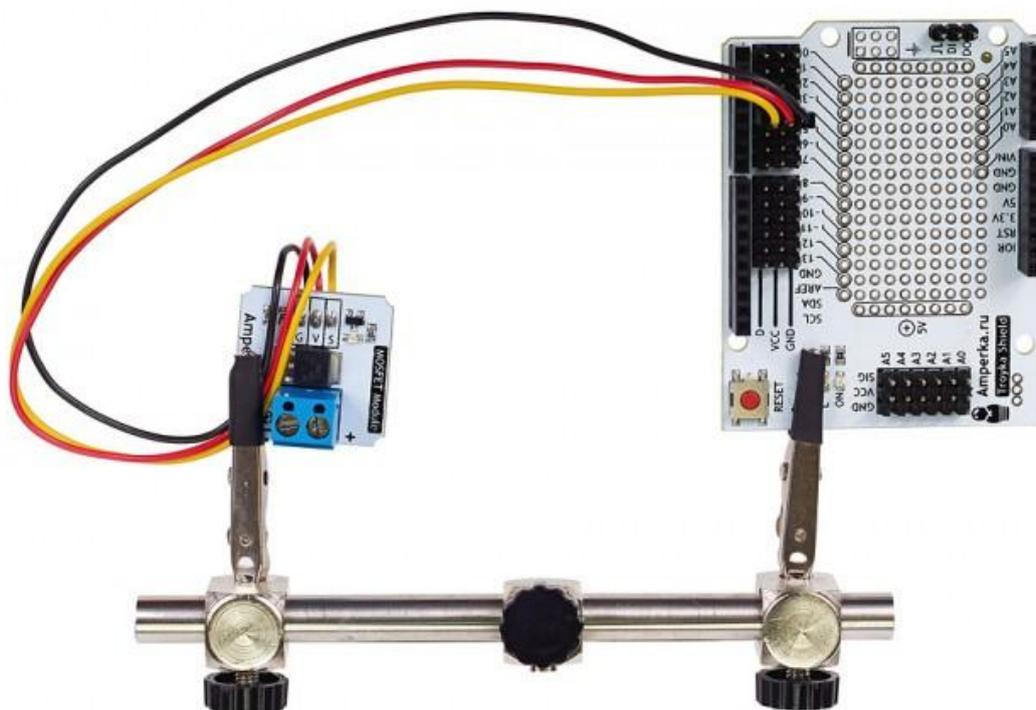
1. Четырёхразрядный индикатор (Тройка-модуль) со шлейфом

Как это собрать?

1. Подключаем дисплей к контакту 3 тройка-шилда. При подключении всех тройка-модулей следите за тем, чтобы чёрный провод соединялся с контактом GND.

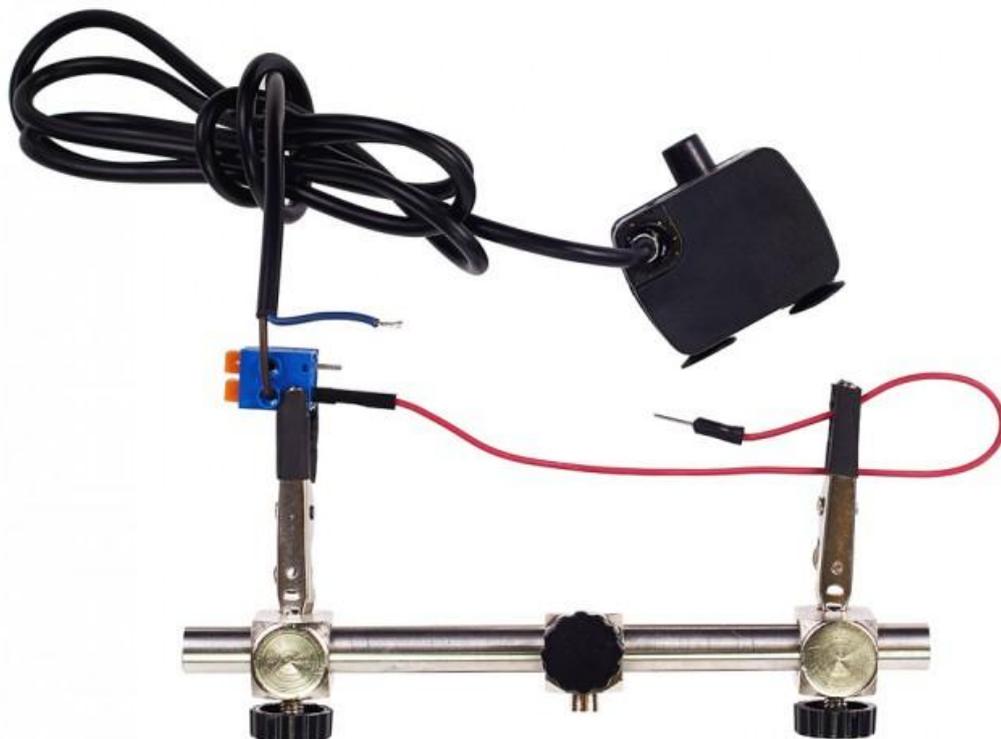


2. Теперь подключаем силовой ключ к контакту 5.

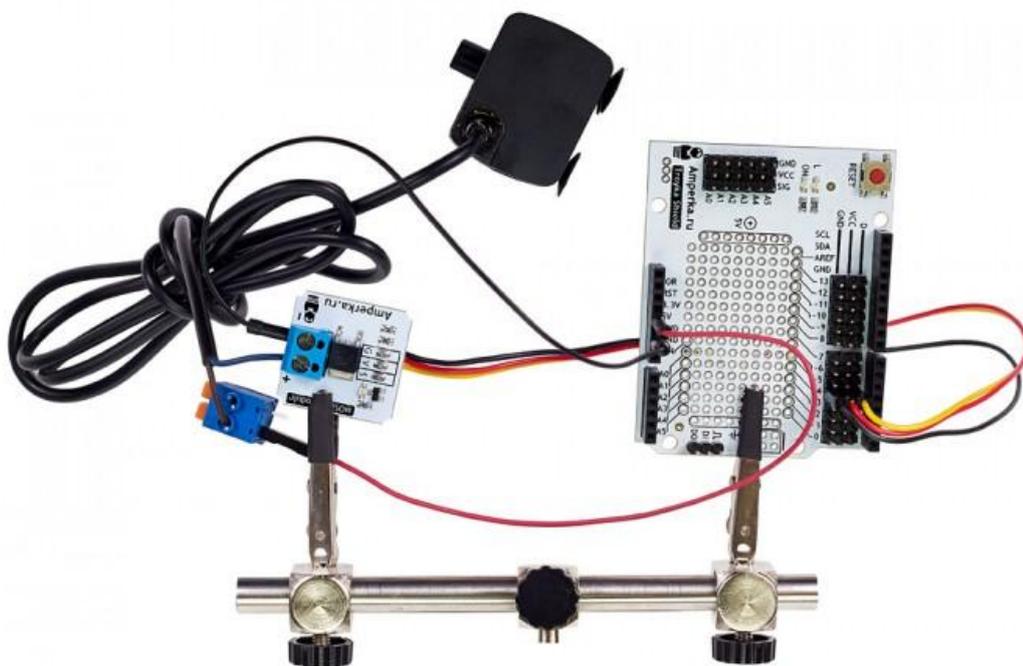


3. У помпы нет контактов на концах проводов, поэтому приходится пользоваться клеммником. Если вы умеете паять, правильной было бы припаять к проводам

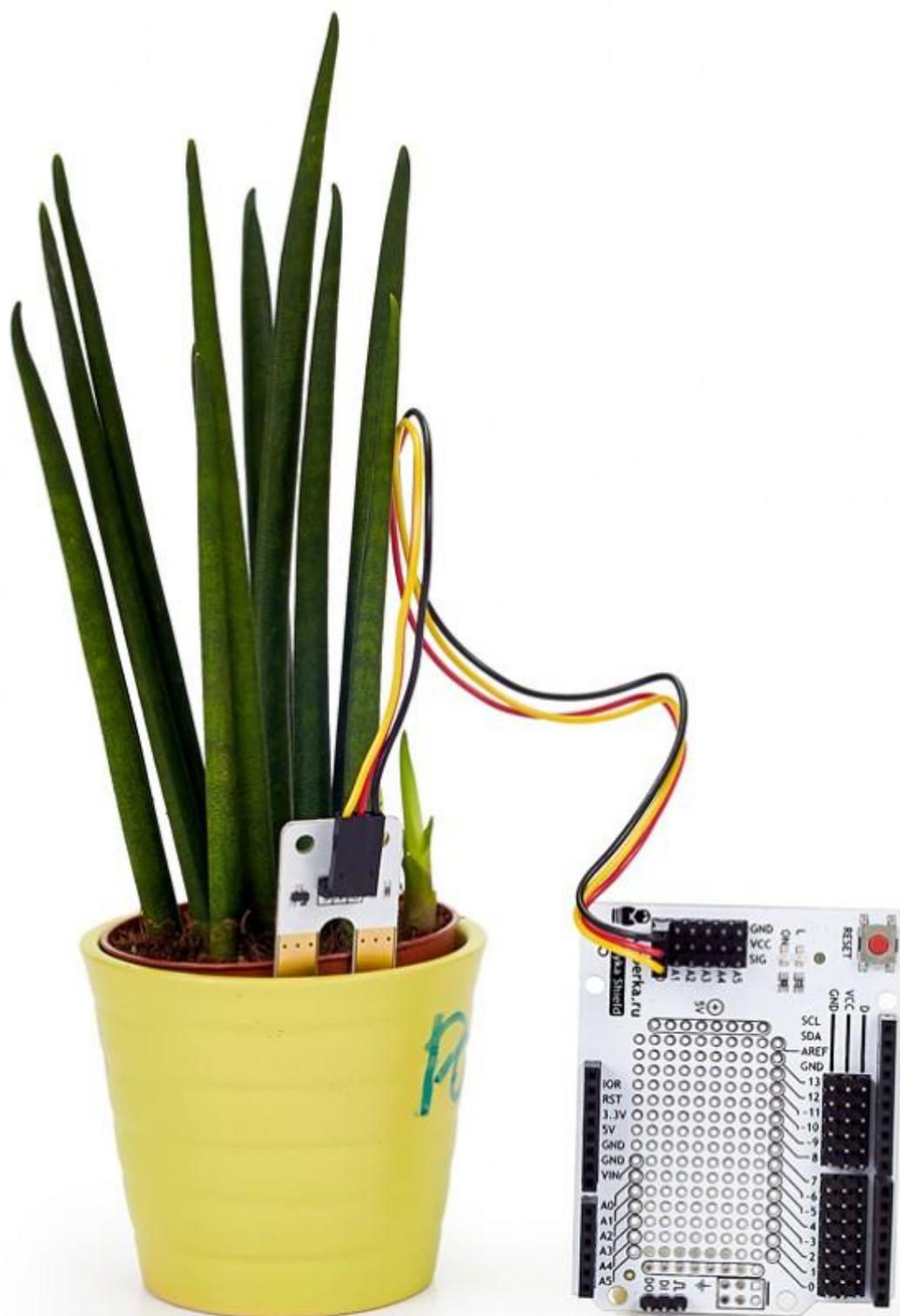
Штырьковые контакты.



4. Полностью подключённое питание помпы выглядит так:



5. Теперь подключаем сенсор влажности почвы.



6. [При помощи Arduino IDE](#) программируем Arduino Uno скетчем [irrigator.ino](#).
7. Втыкаем сенсор влажности в землю.
8. Втыкаем конец трубки с водой в землю. Если горшок с землёй весит менее 2 кг, рекомендуется отдельно закрепить трубку, чтобы она не перевернула растение.
9. Опускаем помпу в ёмкость с водой.
10. Подключаем питание.

Калибровка

Показания датчика влажности сильно зависят от кислотности почвы. Поэтому перед началом пользования ирригатором требуется провести простую процедуру калибровки.

1. Запишите показания на дисплее при воткнутом в сухой горшок сенсоре. Это — минимум влажности.
2. Полейте цветок и дождитесь пока вода полностью впитается в землю и показания сенсора установятся на одном уровне. Запишите их. Это — максимум влажности.
3. В скетче исправьте значения константы HUMIDY_MIN на значение минимальной влажности и HUMIDY_MAX на значение максимальной влажности. Заново прошейте Arduino Uno.

Масштабирование решения

Мы описали решение для одного растения. Но обычно требуется поливать несколько растений. Помимо очевидного решения — подключения к Arduino нескольких помп и датчиков влажности — существует более простое и дешёвое. Достаточно в трубке, которая идёт в комплекте с помпой проделать шилом дырочки на расстоянии около 30 см и воткнуть в эти дырочки куски стержней от обычных шариковых ручек. Выглядеть это будет так:

Горшки с цветами дома часто стоят в ряд на подоконнике. Вам достаточно просто положить трубку на горшки так, чтобы отверстия в ней приходились по одному на горшок. Теперь наше устройство может поливать сразу несколько горшков. Однако в таком случае принимать решение о необходимости полива можно только по одному горшку. Однако обычно горшки примерно одинаковые по размерам и, соответственно, сохнут с примерно равной скоростью. Можно так же комбинировать два решения, разделяя все горшки на группы примерно равных по размерам.

Исходный код

[irrigator.ino](#)

```
#include "QuadDisplay.h"

const int HUMIDY_MIN = 200;
const int HUMIDY_MAX = 700;

void setup(void)
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);

  displayInt(3, 0);
}
```

```

char intDigit(unsigned int val, char pos)
{
    char buf[6];
    sprintf(buf, "%04u", val);
    return buf[3-pos]-'0';
}

unsigned int intDigitSet(unsigned int val, char pos, char d)
{
    char buf[6];
    sprintf(buf, "%04u", val);
    buf[3-pos] = '0'+d;
    return atoi(buf);
}

unsigned int readHumidy(void)
{
    static unsigned int oldval = 0;
    static unsigned int res = 0;
    static unsigned int stability[4] = { 0, 0, 0, 0 };

    char i;
    unsigned int val = analogRead(A0);

    for(i = 0; i < 4; i++) {
        if(intDigit(oldval, i) != intDigit(val, i)) {
            oldval = intDigitSet(oldval, i, intDigit(val, i));
            stability[i] = 0;
        }
        else {
            if(stability[i] < 10) stability[i]++;
            else res = intDigitSet(res, i, intDigit(val, i));
        }
    }

    return res;
}

void loop(void)
{
    static unsigned long wait = millis() + 60000;
    static unsigned int oldhumidy = 0;
    static unsigned long stability = 0;
    unsigned int humidy = readHumidy();

    if(humidy != oldhumidy) {
        oldhumidy = humidy;
        displayInt(3, humidy);
    }

    if(wait != 0 && wait-millis() > 10) {
        return;
    } else wait = 0;

    if(humidy < HUMIDY_MIN) {
        digitalWrite(2, HIGH);
        analogWrite(5, 50);
    }
}

```

```
    delay(1500);  
    digitalWrite(2, LOW);  
    digitalWrite(5, LOW);  
    wait = millis() + 3*60000;  
  }  
}
```