Сенсоры с цифровым сигналом

При использовании цифрового сигнала, сенсор в любой момент времени выдаёт на сигнальный провод либо 0В, либо напряжение своего питания - 5В. Промежуточных значений нет. Для того, чтобы абстрагироваться от конкретных значений напряжения, которые не важны при обработке цифровых сигналов, существуют понятия логического нуля (LOW) и логической единицы (HIGH). 0В - это логический ноль, напряжение питания - это логическая единица.

На **Arduino Uno** имеется 14 цифровых входов, любой из которых может быть использован для подключения такого датчика.

Протокол

Есть простые сенсоры, у которых есть всего два состояния: чёрный/не-чёрный, лево/право и т.д. Их очень легко подключить и считать показания: сенсоры передают свой сигнал непрерывно, а значение на сигнальном проводе напрямую соответствует их показаниям. Такой простой протокол называется бинарным.

Существуют также сенсоры с цифровым сигналом, которые измеряют множество градаций определённой физической величины вроде расстояния или температуры. Но для передачи своих данных с использованием лишь двух значений, каждый такой сенсор определяет собственный протокол. В нём описывается, какие последовательности нулей и единиц, с какими задержками, как несут в себе передаваемые данные. Принимающая сторона, такая как Arduino, должна реализовать алгоритм, который будет считывать показания в соответствии с протоколом. Протокол у каждого сенсора свой, он описывается в его документации. Arduino — очень популярная платформа, поэтому чаще всего реализацию алгоритма расшифровки можно найти в виде готовой библиотеки, написанной энтузиастами или самим производителем сенсора.

Программирование

Если говорить о цифровых сенсорах с бинарным протоколом, считать данные с него - крайне просто.

Поскольку цифровые контакты могут являться как входами, так и выходами, для начала нужно сконфигурировать контакт, к которому подключен сенсор в режим ввода. Это нужно сделать единожды, поэтому функция `setup` - подходящее для этого место. Для конфигурирования режима используется стандартная функция pinMode. Так, например, если вы подключили сенсор к контакту 9, код конфигурации будет выглядеть так:

```
void setup()
{
    pinMode(9, INPUT);
}
```

Затем, чтобы считать состояние в любой момент времени, существует стандартная функция digitalRead. Продолжая пример, чтобы получить состояние сенсора в переменную `value` достаточно исполнить:

```
int value = digitalRead(9);
```

Входное напряжение до 2В проецируется на целочисленное значение 0, что соответствует значению константы LOW; напряжение более 3В проецируется на целочисленное значение 1, что соответствует значению константы HIGH. Напряжение от 2 до 3В спроецируется на 0 или 1 случайным образом, но это не является проблемой, так как цифровые сенсоры не должны выдавать такой сигнал.

Таким образом, программа, которая раз в секунду считывает показания цифрового сенсора с двумя состояниями, подключенного к контакту 9, и посылает их на компьютер может выглядеть так:

digitalSensorRead.pde

```
#define SENSOR_PIN 9

void setup()
{
    pinMode(SENSOR_PIN, INPUT);
    Serial.begin(9600);
}

void loop()
{
    delay(1000);
    int val = digitalRead(SENSOR_PIN);
    Serial.println(val);
}
```

Преимущества и недостатки цифрового сигнала

Преимуществом сенсоров с цифровым сигналом и всего двумя состояниями является крайняя простота их использования с Arduino.

Однако, если речь идёт о цифровом сенсоре с множеством градаций измеряемой величины, их использование с Arduino не так тривиально, как бинарных или аналоговых: необходимо реализовать расшифровку данных, что требует определённых усилий, а также занимает память микроконтроллера.

Поскольку возможных значений в цифровом сигнале всего 2, а возможные отклонения в напряжении «округляются» микроконтроллером в ближайшую сторону, такие сенсоры можно подключать с помощью достаточно длинных (много метров) проводов, не опасаясь искажения сигнала из-за влияния на провод внешних электромагнитных полей.