

Занятие 5 Базовый уровень. Цифровой ввод. Линия

Благодарность слушательнице курса «Основы робототехники» Антроповой Анастасии за помощь в разработке и испытаниях программ для следования по линии

В качестве основы для разработки алгоритмов езды по линии использовалась статья «Следование по линии - руководство по использованию датчиков», статья <https://cxem.net/uprav/uprav40.php>

Цифровой ввод – команда `digitalRead`

Arduino Uno может считывать два уровня напряжения: **+5 В** (*HIGH*, логическая 1) и **0 В** (*LOW*, логический 0) любым своим входом: как цифровыми D0...D13, так и аналоговыми A0...A5

Примечание: используя кодирование информации, можно подать на входы и более сложную информацию, например, с цифрового датчика температуры DS18B20

Для считывания уровня напряжения на пине Arduino используется команда

`digitalRead(номер пина);`

она выдает состояние *HIGH*, если на пине +5 В или состояние *LOW*, если на пине 0 В

Примечание: перед использованием на пине команды `digitalRead` обязательно нужно настроить этот пин в блоке *setup* как входной, используя команду `pinMode(номер пина, INPUT);`

Использование цифрового ввода:

- подключение к контроллеру отдельных кнопок или их набора (создание клавиатуры) для управления и ввода информации
- датчики линии на роботах
- датчики прикосновения на роботах на основе кнопок – «усики» робота
- концевые выключатели на станках с ЧПУ и 3D принтерах, определяющие минимумы и максимумы движения рабочих инструментов по трем осям (X, Y, Z) на основе кнопок или оптических датчиков

Практическое занятие 1. Опрос датчика линии

Нужные компоненты:

- датчик линии
 - провода штырек-разъем: красный провод на +5 В, синий или черный провод (GND), провод любого другого цвета – сигнальный
1. Соберите конструкцию для установки датчика линии на робота по образцу
 2. Подсоедините провода к штырькам датчика линии:
Vcc – красный провод
Gnd – синий или черный провод
D0 – сигнальный провод другого цвета
 3. подключите датчик линии к контроллеру проводами:
5V – красный провод
Gnd – синий или черный провод
D2 – сигнальный провод другого цвета
 4. запустите среду разработки Arduino IDE, откройте пример программы для определения состояния кнопки (нажата или отжата) *Файл / Примеры / Digital / Button*

5. переведем комментарии к данной программе на русский язык:

```
const int buttonPin = 2;    // номер пина, к которому подключена кнопка
const int ledPin = 13;      // светодиод на плате Arduino
int buttonState = 0;        // переменная для хранения статуса кнопки (нажата или не нажата)

void setup() {
  pinMode(ledPin, OUTPUT);   // пин 13 – выходной (управление светодиодом)
  pinMode(buttonPin, INPUT); // пин 2 – входной (чтение кнопки)
}

void loop(){
  buttonState = digitalRead(buttonPin); // читаем состояние кнопки и записываем в переменную
  buttonState

  if (buttonState == HIGH) {           // если в переменной buttonState высокий уровень
    digitalWrite(ledPin, HIGH);        // зажжем светодиод
  }
  else {                               // иначе
    digitalWrite(ledPin, LOW);          // погасим светодиод
  }
}
```

В данной программе используются команды *if*, *else* (если, иначе). Они работают так:

```
if (условие) {
  // выполнить, если условие выполняется
}
else {
  // выполнить, если условие не выполняется
}
```

Примеры условия:

- $x = 10$ – выполняется, если переменная x равна 10
- $x > 10$ – выполняется, если переменная x больше 10
- $x \neq 10$ – выполняется, если переменная x не равна 10

6. Подключите контроллер к компьютеру, загрузите программу *Button* и попробуйте работу: при нажатии/отпускании кнопки светодиод L на плате Arduino должен загораться или потухать

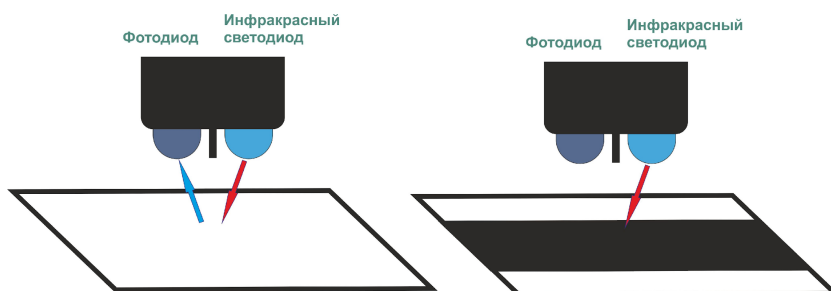
Примечание 1: таким способом можно управлять включением/выключением внешних светодиодов, моторов и т.д.

Примечание 2: опрос кнопок может быть использован в проектах для ввода информации в контроллер (аналогично клавиатуре компьютера) или создания пульта управления по радио (Bluetooth, Wi-Fi и т.п.) роботом.

Использование цифровых датчиков линии для движения по линии

Принцип работы датчика линии:

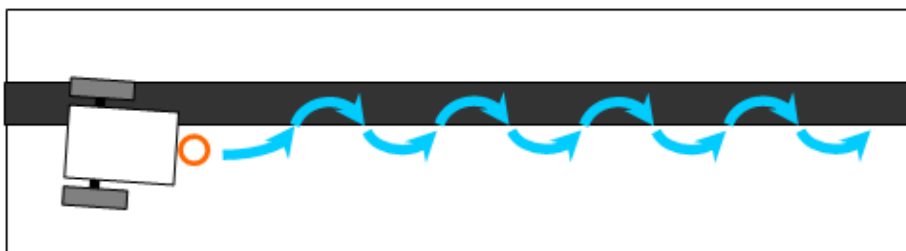
- если свет от ИК светодиода отразился от белого фона и попал на фотодиод – на цифровом выходе датчика одно состояния (например, 0 В, LOW)
- если свет от ИК светодиода не попал на фотодиод – на цифровом выходе датчика другое состояния (например, 5 В, HIGH)



Можно осуществить движение робота с одним таким датчиком по линии (если быть точнее, по границе линии), используя следующий алгоритм:

- Если датчик на светлом фоне – повернуть влево
- Иначе (если датчик на темном фоне) – повернуть вправо

Траектория езды робота при этом показана на рисунке:



Практическое занятие 3. Проверка и настройка датчиков

1. Подключите датчик линии к контроллеру. Контакт VCC к 5V, Gnd к Gnd, D0 к одному из цифровых пинов
2. Откройте пример *Digital / Button*. Замените пин, к которому подключена кнопка (`int buttonPin = 2;`) в примере на тот, к которому подключен датчик линии. Загрузите программу в контроллер
3. Попробуйте датчик над темной и светлой поверхностью. Датчик должен быть слегка поднят над поверхностью, но расстояние от ИК светодиода и приемника (голубой и черный маленькие стеклянные полусферы на датчике) до поверхности – не более 5 мм
4. Если датчик и на светлой, и на темной поверхности реагирует одинаково (состояния светодиодов на датчике и контроллере не меняются), настройте датчик, поворачивая отверткой подстроечные резистор на нем до тех пор, пока состояния светодиодов над темной и светлой поверхностью не будут меняться

По результатам испытаний видим, что когда датчик над светлой поверхностью и светодиод на нем горит, светодиод на контроллере наоборот, погасает. И наоборот, над темной поверхностью светодиод на контроллере загорается. Исходя из загруженной в контроллер программы видно, что **над светлой поверхностью датчик посылает сигнал LOW, а над темной – HIGH**

Общая информация по езде по линии

Так как для всех вариантов, кроме езды на пяти датчиках, будут использоваться только пины для вращения моторов вперед, можно прописывать только их. Пример для езды вперед:

```
analogWrite (LF, speed_left);  
analogWrite (RF, speed_right);
```

Так как моторы на роботе могут быть разными, задаются значения скоростей отдельно для левого мотора, отдельно для правого. При этом к скоростям могут добавляться: *soft* – для слабого поворота, *hard* – быстрый поворот, *very_hard* – очень быстрый поворот

Ситуация, когда датчик на черной линии – логическая 1, когда на белом поле – логический 0

Опрос датчиков линии (цифровой выход) происходит аналогично опросу кнопки – *Файл / Примеры / Digital / Button*. Для того, чтобы избежать путаницы в состояниях, определим точно, какое состояние соответствует черному (в нашем случае –высокий уровень HIGH), какое – белому (низкий уровень LOW) под датчиком и проведем следующее определение:

```
#define BLACK HIGH  
#define WHITE LOW
```

Данные определения пишутся там же, где все переменные – до блока *setup*. Обратите внимание, что в данном случае ; на конце не ставится. Данная команда определяет для контроллера, что BLACK – то же самое, что HIGH, а WHITE – то же самое, что LOW. Например, после такого определения строки:

```
if (left_pin == HIGH)  
if (left_pin == BLACK)
```

будут одинаковы. Но работа с большим количеством условий удобнее, когда однозначно определены состояния датчиков – на черной линии BLACK или на белом поле WHITE.

Движение по линии на одном датчике

Примечание: значения скоростей здесь и далее даны ориентировочно, их нужно подбирать самостоятельно для своего конкретного робота

Вариант	Ситуация	Что делать	Названия и примеры значений скоростей
0	датчик на белом поле	ехать вправо	speed_left = 150 0
1	датчик на линии	ехать влево	0 speed_right = 150

Примечание: возможно, здесь и далее придется сделать вариант вперед-влево и вперед-вправо. В этом случае на один мотор надо подавать бОльшую скорость, на другой (где был 0 – меньшую)

Движение по линии на двух датчиках

Рассматривается вариант, когда датчики находятся на расстоянии бОльшем ширины линии

Вариант	Ситуация	Что делать	Названия и примеры значений скоростей
0 0	оба датчика на белом поле	ехать прямо	speed_left = 150 speed_right = 150
1 0	левый датчик на линии	ехать вперед влево	speed_left_soft = 25 speed_right = 150
0 1	правый датчик на линии	ехать вперед вправо	speed_left = 150 speed_right_soft = 25
1 1	оба датчика на линии	–	этот вариант не учитывается

Движение по линии на трех датчиках

Третий датчик используется для определения съезда с линии:

два крайних датчика при срабатывании на линии немного корректируют положение робота на линии

центральный датчик при срабатывании вне линии дает команду использовать большую скорость для возврата на линию (направление приложения определяется по тому, какой из крайних датчиков сработал)

Вариант	Ситуация	Что делать	Названия и примеры значений скоростей
0 0 0	все датчики на белом поле	робот сошел с линии	этот вариант не учитывается
0 1 0	центральный датчик на линии, крайние на поле	ехать прямо	speed_left = 150 speed_right = 150
1 1 0	робот немного сдвинут влево	повернуть немного вправо	speed_left = 150 speed_right_soft = 40
0 1 1	робот немного сдвинут вправо	повернуть немного влево	speed_left_soft = 40 speed_right = 150
1 0 0	робот сильно сдвинут влево	повернуть сильно вправо	speed_left = 150 speed_right_hard = 0
0 0 1	робот сильно сдвинут вправо	повернуть сильно влево	speed_left_hard = 0 speed_right = 150
1 1 1	все датчики на линии	робот на перекрестке	этот вариант не учитывается

Движение по линии на пяти датчиках

Добавляются еще два датчика по краям, срабатывание одного из которых показывает, что робот очень сильно сошел с линии и необходимо приложить сильное воздействие, чтобы вернуть его обратно. Вплоть до разворота на месте (то есть подачи сигналов на пины вращения моторов назад). Таким образом, везде придется прописывать состояния для всех четыре пинов драйвера. Пример для медленного разворота

```
analogWrite (LF, speed_left);
analogWrite (LB, 0);
analogWrite (RF, speed_right);
analogWrite (RB, 0);
```

Вариант	Ситуация	Что делать	Названия и примеры значений скоростей
0 0 0 0 0	все датчики на белом поле	робот сошел с линии	этот вариант не учитывается
0 0 1 0 0	центральный датчик на линии, крайние на поле	ехать вперед	speed_left = 150 speed_right = 150
0 1 1 0 0	слегка слева	повернуть слегка влево	speed_left_soft = 150 speed_right = 80
0 0 1 1 0	слегка справа	повернуть слегка вправо	speed_left_soft = 80 speed_right = 150

0 1 0 0 0	средне слева	повернуть налево с нормальной скоростью	speed_left_normal = 50 speed_right = 150
0 0 0 1 0	средне справа	повернуть направо с нормальной скоростью	speed_left = 150 speed_right_normal = 50
1 1 0 0 0	сильно слева	повернуть налево сильно	speed_left_hard = 10 speed_right = 150
0 0 0 1 1	сильно справа	повернуть направо сильно	speed_left = 150 speed_right_hard = 10
1 0 0 0 0	очень сильно слева	развернуться налево	speed_left_back = 30 (на LB) speed_right = 150
0 0 0 0 1	очень сильно справа	развернуться направо	speed_left = 150 speed_right_back = 30 (на RB)
1 1 1 1 1	все датчики на линии	робот на перекрестке	этот вариант не учитывается