Тема 7 · Урок Ф.1

Энкодеры: считаем обороты и точное расстояние

Робот считает обороты колёс и проезжает отмеренное расстояние, а не «сколько успеет за время».

2 × 45 мин | 1–2 ученика на робота | Практика

РЕЗУЛЬТАТ УРОКА

Программа считает обороты колёс энкодерами и проезжает отмеренное расстояние. Число импульсов на оборот измерено, точность проверена линейкой. Не нужен для итоговой задачи.

Ты научишься

- Объяснить, как энкодер превращает оборот колеса в импульсы.
- Измерить число импульсов на оборот колеса.
- Пересчитать расстояние в импульсы через длину окружности колеса.
- Проехать отмеренное расстояние и оценить точность линейкой.

Нужно знать

- 2.1 Моторы: едем и останавливаемся
- 3.2 П-регулятор: плавно и быстро

Понадобится

Колесо × 2 | Мотор-редуктор × 2 | USB-кабель для Arduino

Зачем этот урок

В уроке 2.1 робот ехал «по времени»: моторы включались на две секунды. Но сколько он проедет за две секунды, зависит от заряда батарей, пола и веса. **Энкодер** считает обороты колеса, поэтому программа может выбирать момент остановки по вращению колёс, а не только по времени. Точность пути всё равно нужно проверить линейкой.

Для урока нужен дополнительный модуль: энкодер на каждое колесо. В основной комплект курса он не входит; без него урок не провести. Факультатив не нужен для итоговой задачи.

РАЗМИНКА · ВСПОМНИ

1. Как в уроке 2.1 робот проезжал заданный участок?
2. Почему при тех же двух секундах робот с севшими батареями проезжает меньше?
3. Что такое П-регулятор из урока 3.2 в одной фразе?

ЗАНЯТИЕ 1 · 45 МИН

1 Считаем импульсы

1 Почему время — плохая линейка

8 мин

Что должен делать робот. Роботу нужно проехать заданное расстояние точнее, чем по секундомеру. Для этого понадобятся дополнительные энкодеры: в базовом комплекте их нет. Они измеряют вращение колёс, а не расстояние до препятствия.

Что используем и как это работает. Энкодер выдаёт импульсы при вращении. Счётчик накапливает их; число импульсов за оборот связывает счёт с оборотами колеса. Длина окружности связывает обороты с предполагаемым путём. Прерывание позволяет быстро увеличивать счётчик при импульсе, даже когда основная программа занята другой работой.

Как связать измерение с действием. Измерим импульсы на оборот и используем диаметр своего колеса. На AVR чтение многобайтного счётчика прерывание может разорвать посередине, поэтому копируем его при кратко выключенных прерываниях. Проверим результат линейкой: при проскальзывании колесо вращается, а корпус проходит меньше, поэтому энкодер не даёт абсолютной координаты.

1. Измерь пути «по времени» при скорости 100, 150 и 200. Построй в журнале таблицу «скорость — путь».
2. Оцени: можно ли по этой таблице пересчитать время для 50 см на скорости 120? Насколько ты доверяешь такому пересчёту завтра, на другом заряде батарей? Объясни.
3. Запиши, что нужно **измерять** во время езды, чтобы путь не зависел от этих причин.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Таблица, оценка пересчёта и ответ — в журнале.

2 Как устроен энкодер

12 мин

1. Посчитай прорези, запиши контакты энкодеров из таблицы проводки.
2. Оцени частоту импульсов: колесо 45 мм, 20 прорезей, робот едет 0,3 м/с. Сколько импульсов в секунду даёт каждый энкодер?
3. Объясни в журнале, почему счёт делают прерыванием и почему на Arduino Uno для этого подходят только контакты 2 и 3.

⚠ **БЕЗОПАСНОСТЬ**

Не подключай энкодеры при включённом питании. Контакты не указаны в таблице — подключает преподаватель.

Подсказка: Частота импульсов

Сначала — сколько оборотов в секунду делает колесо при 300 мм/с.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Оценка частоты и объяснение — в журнале.

3 Импульсов на один оборот

22 мин

Требование 1. Программа считает импульсы каждого колеса прерыванием и печатает оба счётчика в монитор порта; счётчики читаются так, что прерывание не портит число. Без вписанных контактов программа не идёт дальше.

1. Напиши программу сам.

Как это устроено: Справка по командам

Команда	Что делает
`attachInterrupt(digitalPinToInterrupt(pin), f, RISING)`	Вызывать функцию `f` на каждый фронт импульса на контакте `pin`
`volatile long x`	Переменная, которую меняет прерывание
`noInterrupts()` / `interrupts()`	Выключить и снова включить прерывания

2. Измерь импульсы на оборот: прокрути колесо 10 оборотов по метке, потом **20 оборотов**. Насколько различаются два результата? Какой точнее и почему?
3. Найди положение колеса, при котором счёт растёт без движения. Объясни в журнале, что происходит в датчике.

Подсказка: Идея (1 из 2)

Функции прерывания должны быть короткими: только увеличить счётчик.

Подсказка: Счёт без движения (2 из 2)

На краю прорези свет попадает на приёмник «наполовину», и сигнал дрожит около порога.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Программа выполняет требование 1, импульсы на оборот измерены дважды, объяснение дрожания записано.

2

ЗАНЯТИЕ 2 · 45 МИН

Точное расстояние

4 Миллиметры в импульсы

15 мин

Требование 2. Функция `long ticksFor(int mm, int ticksPerRev, int diameterMm)` переводит миллиметры в импульсы с округлением до ближайшего целого и не переполняется на Arduino Uno.

Приёмка. Напиши функцию в тестовом скетче. Когда все строки `PASS`, перенеси её в свою программу.

Тест функции ticksFor LF1_Ticks_Test.ino

```
// Проверка функции ticksFor. Напиши функцию здесь, загрузи и открой монитор
// Когда все строки PASS, перенеси ticksFor в свою программу.

// Требование 2: сколько импульсов нужно, чтобы проехать mm миллиметров (окру
long ticksFor(int mm, int ticksPerRev, int diameterMm) {
  return 0;
}

int failed = 0;

void expect(int mm, int ticksPerRev, int diameterMm, long expected) {
  long actual = ticksFor(mm, ticksPerRev, diameterMm);
  Serial.print(actual == expected ? "PASS " : "FAIL ");
  Serial.print("ticksFor(");
  Serial.print(mm);
  Serial.print(", ");
  Serial.print(ticksPerRev);
  Serial.print(", ");
  Serial.print(diameterMm);
  Serial.print(") = ");
  Serial.print(actual);
  Serial.print(", ожидаем ");
  Serial.println(expected);
  if (actual != expected) {
    failed++;
  }
}

void setup() {
  Serial.begin(9600);

  expect(0, 20, 45, 0);
  expect(141, 20, 45, 20);   // почти ровно один оборот
  expect(500, 20, 45, 71);   // 70.7 округляется вверх
  expect(1000, 20, 65, 98);  // другое колесо
  expect(1000, 40, 45, 283);
  expect(2000, 40, 45, 566); // 2000 × 40 не помещается в int

  if (failed == 0) {
    Serial.println("Все проверки пройдены");
  } else {
```

```
    Serial.print("Ошибок: ");
    Serial.println(failed);
  }
}

void loop() {
}
```

Требование 3. Программа едет вперёд, пока **среднее** двух счётчиков не достигнет цели, останавливается и печатает цель и насчитанное после остановки. Без вписанного числа импульсов на оборот робот стоит.

1. Выполни требования 2 и 3, проедь 500 мм от метки и измерь путь.

> **Подсказка: Идея (1 из 2)**
>
> Импульсы = обороты × импульсы на оборот, обороты = путь / длина окружности.

> **Подсказка: Тест на 2000 мм не проходит (2 из 2)**
>
> `2000 * 40` — 80 000, больше, чем вмещает `int`. Считай в `float` или `long` до умножения.

> **ДОЛЖНО ПОЛУЧИТЬСЯ**
> Тест пройден, поездка на 500 мм сделана и измерена.

5 Почему робот проезжает дальше

15 мин

Требование 4. За `SLOW_TICKS` импульсов до цели робот сбавляет скорость, чтобы меньше катиться после остановки.

1. Выполни требование 4 и подбери `SLOW_TICKS` и пониженную скорость.
2. Сравни три заезда без замедления и три с замедлением: среднюю ошибку и разброс.
3. Объясни в журнале, чем замедление лучше поправки «остановиться раньше на N импульсов».

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Требование 4 выполнено, сравнение двух способов и вывод — в журнале.

6 Задания по уровням

12 мин

★★ **ПРОДВИНУТЫЙ УРОВЕНЬ · РОВНО 90 ГРАДУСОВ**

Повернуть на месте на 90°: колёса крутятся в разные стороны, каждое проходит четверть окружности диаметром в колею робота — 90 мм. Посчитать импульсы, проверить угол угольником. Ошибка — не больше 10°.

ПРОВЕРЬ СЕБЯ

- [ ] Я могу объяснить, как энкодер считает обороты.
- [ ] Я могу перевести миллиметры в импульсы.
- [ ] Я могу объяснить, почему робот проезжает чуть дальше цели и как это уменьшить.

Проверь себя

★★ Продвинутый уровень

1. **Робот поворачивает на месте на 90°, колея 90 мм. Какой путь проходит каждое колесо?**

- ○ Около 71 мм: четверть окружности диаметром 90 мм
- ○ Около 141 мм
- ○ 90 мм

2. **Функция верна на 500 мм, но для 2000 мм и 40 импульсов на оборот даёт неверное число. В чём ошибка?**

```
long ticksFor(int mm, int ticksPerRev, int diameterMm) {
  int ticks = mm * ticksPerRev / (3.14 * diameterMm);
  return ticks;
}
```

- ○ 2000 × 40 = 80 000 не помещается в int Arduino Uno
- ○ Нельзя делить на 3.14
- ○ Функция должна возвращать int

3. **Колесо стоит на краю прорези, а счётчик растёт. Что происходит?**

- ○ Сигнал дрожит около порога, и датчик даёт лишние импульсы
- ○ Колесо незаметно крутится
- ○ Прерывание срабатывает само по таймеру

4. Почему цель сравнивают со средним двух счётчиков, а не с одним левым?

- ○ Колёса крутятся немного по-разному; среднее ближе к пути середины робота
- ○ Левый энкодер всегда врёт
- ○ Так программа работает быстрее

5. Счётчик long читают в loop() без noInterrupts(). Что может случиться?

- ○ Прерывание изменит число посередине чтения, и программа увидит смесь старого и нового значения
- ○ Ничего: Arduino читает long за один шаг
- ○ Энкодер перестанет считать

Отвечай на вопросы своего уровня. Нужно верных ответов: Начальный — 4 из 5; Средний — 4 из 5; Продвинутый — 4 из 5. Ответы проверит преподаватель или программа курса.

Итог

Энкодер превращает оборот колеса в импульсы, а импульсы — в пройденный путь. Робот едет не «сколько успеет», а сколько нужно, при любом заряде батарей.

Как подтвердить результат

Результат опыта связывает условия, версию программы, наблюдение и вывод. Сравнение попыток с одним изменением помогает проверить причину, включая неудачные попытки. Отметка относится к уже выполненной практике; повтор ради отметки не нужен. Расхождение с прогнозом задаёт следующую проверяемую гипотезу. Модель, измерения настоящего робота и объяснение — разные виды доказательств.

Словарь урока

энкодер	Датчик оборотов: диск с прорезями на оси колеса и датчик, который даёт импульс на каждую прорезь.
импульс	Короткий сигнал энкодера: прорезь прошла мимо датчика.
прерывание	Arduino откладывает основную программу и сразу выполняет короткую функцию, когда на контакт пришёл импульс.
volatile	Пометка переменной, которую меняет прерывание: программа каждый раз читает её заново.
длина окружности	Путь колеса за один оборот: π × диаметр.