

Приложение Б — листинги программного обеспечения контроллера управления

Листинг 1. Алгоритм управления шаговым двигателем

```
//Первый двигатель
const uint8_t pin_ENA1 = 2;           // Вывод Arduino подключённый к входу драйвера
ENA+.
const uint8_t pin_DIR1 = 3;           // Вывод Arduino подключённый к входу драйвера DIR+.
const uint8_t pin_PUL1 = 4;          // Вывод Arduino подключённый к входу драйвера
PUL+.
//Второй двигатель
const uint8_t pin_ENA2 = 8;           // Вывод Arduino подключённый к входу драйвера
ENA+.
const uint8_t pin_DIR2 = 9;           // Вывод Arduino подключённый к входу драйвера DIR+.
const uint8_t pin_PUL2 = 10;          // Вывод Arduino подключённый к входу драйвера
PUL+.
//Третий двигатель
const uint8_t pin_ENA3 = 5;           // Вывод Arduino подключённый к входу драйвера
ENA+.
const uint8_t pin_DIR3 = 6;           // Вывод Arduino подключённый к входу драйвера DIR+.
const uint8_t pin_PUL3 = 7;          // Вывод Arduino подключённый к входу драйвера
PUL+.

uint32_t f = 1;    // Определяем частоту следования микрошагов от 1 до 200'000 Гц.
                // Чем выше частота, тем выше скорость вращения вала.
void setup() {
    pinMode( pin_ENA1, OUTPUT );       // Конфигурируем выводы Arduino как выходы для
драйвера первого двигателя.
    pinMode( pin_DIR1, OUTPUT );
    pinMode( pin_PUL1, OUTPUT );

    pinMode( pin_ENA2, OUTPUT );       // Конфигурируем выводы Arduino как выходы для
драйвера второго двигателя.
    pinMode( pin_DIR2, OUTPUT );
    pinMode( pin_PUL2, OUTPUT );

    pinMode( pin_ENA3, OUTPUT );       // Конфигурируем выводы Arduino как выходы для
драйвера третьего двигателя.
    pinMode( pin_DIR3, OUTPUT );
    pinMode( pin_PUL3, OUTPUT );

    pinMode(LED_BUILTIN, OUTPUT);      // Конфигурируем вывод Arduino как выход для
светодиода.
}

#define t 20    // Определяем длительность импульсов t3 и пауз t4 в мкс.
```

```

void loop() {    // рабочий цикл

                                // движение вала первого двигателя.
    digitalWrite( pin_ENA1, 0 );    // Разрешаем работу двигателя.
    delay(5);                      // Выполняем задержку t1 (см. график STEP/DIR).
    digitalWrite( pin_DIR1, 1);    // Выбираем направление вращения.
    delay(5);                      // Выполняем задержку t2 (см. график STEP/DIR).
    // Поворачиваем вал на 2 оборота:
    for (int i = 0; i < 100; i++) {    // Выполняем 1600 проходов цикла (1 оборот = 800 тактов, 2
    оборота = 1600).
        digitalWrite( pin_PUL1, 1 );
        digitalWrite(LED_BUILTIN, HIGH);    // Устанавливаем на выводе PUL состояние
логической «1».
        delay(t);                      // Выполняем задержку t3 (см. график STEP/DIR).
        digitalWrite( pin_PUL1, 0 );
        digitalWrite(LED_BUILTIN, LOW);
        delay(t);                      // Выполняем задержку t4 (см. график STEP/DIR).
    }

    // Останавливаем вал удерживая его (Двигатель не вращается, если на вывод PUL не
    поступают импульсы.):
    delay(100);                      // Ждём 5 секунд. В это время двигатель остановлен, его вал
    удерживается.

    // Меняем направление движения вала:
    digitalWrite( pin_DIR1, 0 );    // Меняем логический уровень вывода DIR с 0 на 1.
    delay(5);                      // Выполняем задержку t2 (см. график STEP/DIR).

    // Поворачиваем вал на 2 оборота:
    for (int i = 0; i < 100; i++) {    // Выполняем 1600 проходов цикла (1 оборот = 800 тактов, 2
    оборота = 1600).
        digitalWrite( pin_PUL1, 1 );    // Устанавливаем на выводе PUL состояние логической
«1».
        delay(t);                      // Выполняем задержку t3 (см. график STEP/DIR).
        digitalWrite( pin_PUL1, 0 );    // Устанавливаем на выводе PUL состояние логического
«0».
        delay(t);                      // Выполняем задержку t4 (см. график STEP/DIR).
    }

    // Останавливаем вал без удержания:
    digitalWrite( pin_ENA1, 1 );    // Запрещаем работу двигателя, отключаем токи в
    обмотках.
    delay(1000);                    // Ждём 5 секунд. В это время двигатель отключен, его вал
    свободен.

```

```

// движение вала второго двигателя.
digitalWrite( pin_ENA2, 0 );    // Разрешаем работу двигателя.
delay(5);                      // Выполняем задержку t1 (см. график STEP/DIR).
digitalWrite( pin_DIR2, 1);     // Выбираем направление вращения.
delay(5);                      // Выполняем задержку t2 (см. график STEP/DIR).
// Поворачиваем вал на 2 оборота:
for (int i = 0; i < 100; i++) { // Выполняем 1600 проходов цикла (1 оборот = 800 тактов, 2
оборот = 1600).
    digitalWrite( pin_PUL2, 1 );
    digitalWrite(LED_BUILTIN, HIGH);    // Устанавливаем на выводе PUL состояние
логической «1».
    delay(t);                          // Выполняем задержку t3 (см. график STEP/DIR).
    digitalWrite( pin_PUL2, 0 );
    digitalWrite(LED_BUILTIN, LOW);
    delay(t);                          // Выполняем задержку t4 (см. график STEP/DIR).
}
// Останавливаем вал удерживая его (Двигатель не вращается, если на вывод PUL не
поступают импульсы):
delay(100);                        // Ждём 5 секунд. В это время двигатель остановлен, его вал
удерживается.

// Меняем направление движения вала:
digitalWrite( pin_DIR2, 0 );    // Меняем логический уровень вывода DIR с 0 на 1.
delay(5);                      // Выполняем задержку t2 (см. график STEP/DIR).

// Поворачиваем вал на 2 оборота:
for (int i = 0; i < 100; i++) { // Выполняем 1600 проходов цикла (1 оборот = 800 тактов, 2
оборот = 1600).
    digitalWrite( pin_PUL2, 1 );    // Устанавливаем на выводе PUL состояние логической
«1».
    delay(t);                      // Выполняем задержку t3 (см. график STEP/DIR).
    digitalWrite( pin_PUL2, 0 );    // Устанавливаем на выводе PUL состояние логического
«0».
    delay(t);                      // Выполняем задержку t4 (см. график STEP/DIR).
}

// Останавливаем вал без удержания:
digitalWrite( pin_ENA2, 1 );    // Запрещаем работу двигателя, отключаем токи в
обмотках.
delay(1000);                    // Ждём 5 секунд. В это время двигатель отключен, его вал
свободен.

// движение вала третьего двигателя.
digitalWrite( pin_ENA3, 0 );    // Разрешаем работу двигателя.
delay(5);                      // Выполняем задержку t1 (см. график STEP/DIR).
digitalWrite( pin_DIR3, 1);     // Выбираем направление вращения.

```

```

delay(5);                // Выполняем задержку t2 (см. график STEP/DIR).
// Поворачиваем вал на 2 оборота:
for (int i = 0; i < 100; i++) {    // Выполняем 1600 проходов цикла (1 оборот = 800 тактов, 2
оборот = 1600).
    digitalWrite( pin_PUL3, 1 );
    digitalWrite(LED_BUILTIN, HIGH);    // Устанавливаем на выводе PUL состояние
логической «1».
    delay(t);                // Выполняем задержку t3 (см. график STEP/DIR).
    digitalWrite( pin_PUL3, 0 );
    digitalWrite(LED_BUILTIN, LOW);
    delay(t);                // Выполняем задержку t4 (см. график STEP/DIR).
}
// Останавливаем вал удерживая его (Двигатель не вращается, если на вывод PUL не
поступают импульсы.):
delay(100);                // Ждём 5 секунд. В это время двигатель остановлен, его вал
удерживается.

// Меняем направление движения вала:
digitalWrite( pin_DIR3, 0 );    // Меняем логический уровень вывода DIR с 0 на 1.
delay(5);                // Выполняем задержку t2 (см. график STEP/DIR).

// Поворачиваем вал на 2 оборота:
for (int i = 0; i < 100; i++) {    // Выполняем 1600 проходов цикла (1 оборот = 800 тактов, 2
оборот = 1600).
    digitalWrite( pin_PUL3, 1 );    // Устанавливаем на выводе PUL состояние логической
«1».
    delay(t);                // Выполняем задержку t3 (см. график STEP/DIR).
    digitalWrite( pin_PUL3, 0 );    // Устанавливаем на выводе PUL состояние логического
«0».
    delay(t);                // Выполняем задержку t4 (см. график STEP/DIR).
}

// Останавливаем вал без удержания:
digitalWrite( pin_ENA3, 1 );    // Запрещаем работу двигателя, отключаем токи в
обмотках.
delay(1000);                // Ждём 5 секунд. В это время двигатель отключен, его вал
свободен.
}

```

Листинг 2. Процедура обработки прерывания от таймера 1. Фильтрация сигнала с линейной шкалы

```
int a = 0;
void TimerCallback()
{
    if (length1 > 45)
    {
        length1 = 45;
    }
    if (length1 < -45)
    {
        length1 = -45;
    }
    FirFracE.x = length1<<7;
    FirFrac15Calc5(&FirFracE);
    se = (FirFracE.y>>7)+45;

    if (length2 > 128)
    {
        length2 = 128;
    }
    if (length2 < -128)
    {
        length2 = -128;
    }

    FirFracA.x = length2<<7;
    FirFrac15Calc5(&FirFracA);
    sa = (FirFracA.y>>5) + 1600;

    if (length3 > 128)
    {
        length3 = 128;
    }
    if (length3 < -128)
    {
        length3 = -128;
    }

    FirFracB.x = length3<<7;
    FirFrac15Calc5(&FirFracB);
    sb = (FirFracB.y>>6) + 1450;
    if (sb < 1000)
    {
        sb = 1000;
    }
    if (sb > 1450)
    {
```

```
    sb = 1450;
}

TCCR1A |= _BV(COM1A1);
TCCR1A |= _BV(COM1B1);
pinMode(TIMER1_A_PIN, OUTPUT);
pinMode(TIMER1_B_PIN, OUTPUT);
OCR1A = sa;
OCR1B = sb;
if (++a >= 50)
{
    a = 0;
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
}
}
```