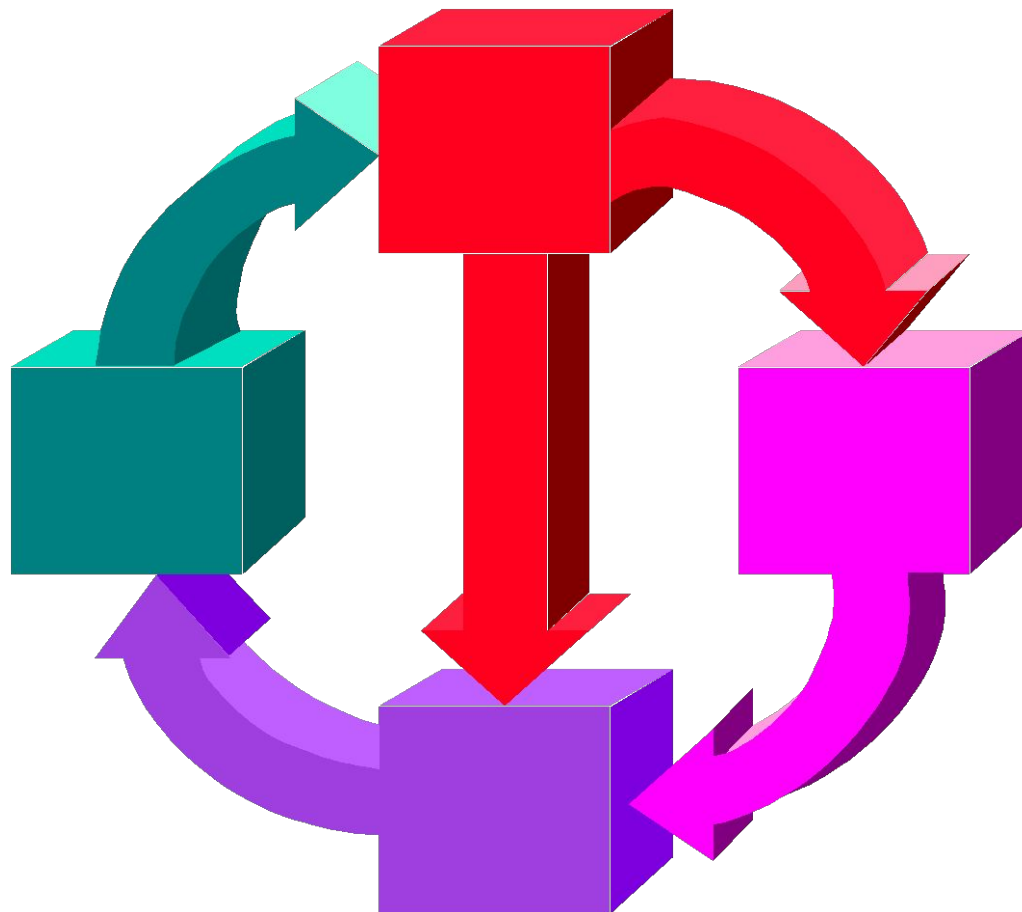


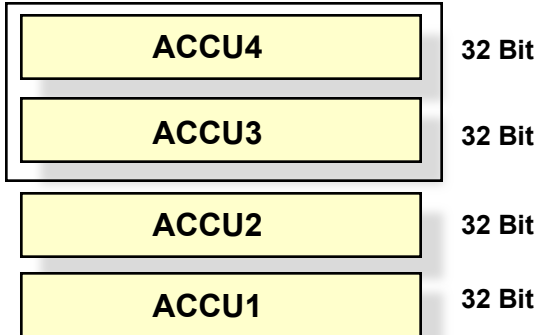
Инструкции, зависящие от битов слова статуса



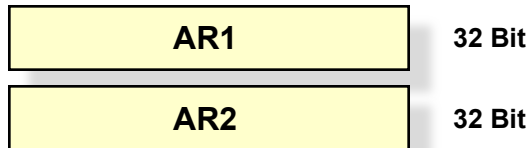
Регистры и области памяти S7-CPU

Аккумуляторы

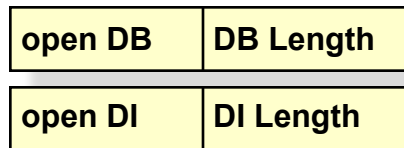
Только в S7-400



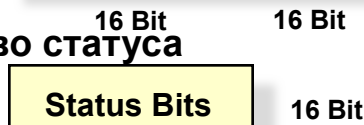
Адресные регистры



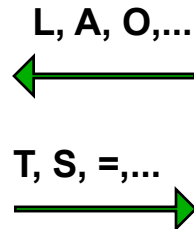
Регистры блоков данных



Слово статуса



Регистры S7-CPU



Загрузочная память

- Логические блоки
- Блоки данных

Рабочая память

- Логические блоки
- Блоки данных

Области ввода/вывода

Системная память

- Табл.отобр.входн. процесса
- Табл.отобр.выходн. процесса - Меркеры
- Таймеры
- Счетчики
- Локальный стек
- Стек прерываний
- Стек блоков

Области памяти S7-CPU

Структура слова статуса

Значение битов в слове статуса

Бит	Название	Величина	Значение
0	/FC	2^0	Бит первичного опроса
1	RLO	2^1	Результат логической операции
2	STA	2^2	Статус
3	OR	2^3	Или
4	OS	2^4	Переполнение с запоминанием
5	OV	2^5	Переполнение
6	CC 0	2^6	Код состояния
7	CC 1	2^7	Код состояния
8	BR	2^8	Двоичный результат
9...15	Не используются		$2^9 \dots 2^{10}$

Проверка битов слова статуса

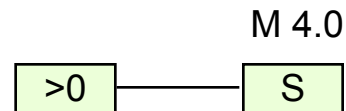
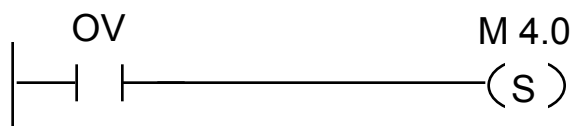
Проверка в STL

- A OV Просмотр переполнения
- A OS Просмотр переполнения с запоминанием
- A BR Просмотр BR-флага

Проверка кода состояния (CC0, CC1)

- A == 0 Результат равен 0
- A > 0 Результат больше 0
- A <> 0 Результат не равен 0
- A =< 0 Результат меньше либо равен 0
и т.д.
- A UO Потеря порядка

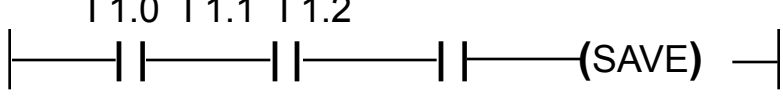
Проверка в LAD и в FBD

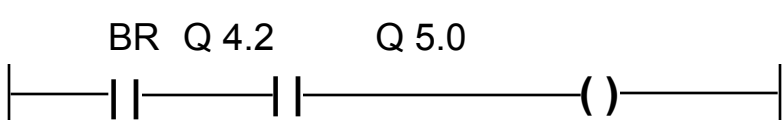


Инструкции с битами слова статуса

Инструкция Значение Пример

- SET Установка RLO в "1" SET //RLO-1
= M 0.1
- CLR Установка RLO в "0" CLR //RLO-0
- NOT Инвертирование RLO O Manual_operation
O Automatic_operation
NOT
= Operating_modes
= M0.0

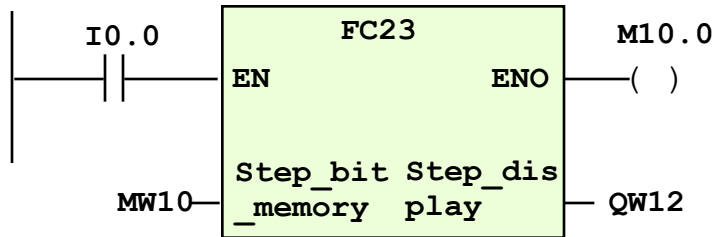
- SAVE Сохранить RLO в BR
(в бинарном результате) 

- A BR Опросить BR 

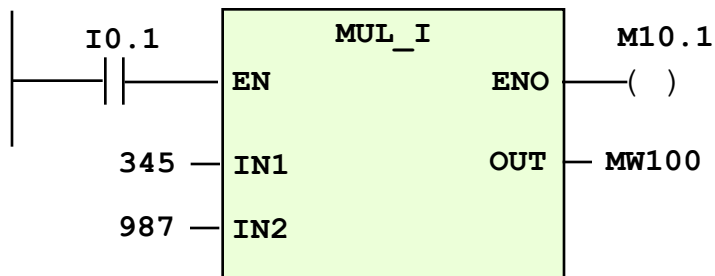
BR бит и ENO в вызовах блока и в сложных функциях

LAD

Network 1: Cyclic Program



Network 2: ???



STL

Network 1: Cyclic Program

```

A    I    0.0
JNB  _001
CALL FC   23
      Step_bit_memory    :=MW10
      Step_display      :=QW12
_001:  A    BR
      =    M    10.0
    
```

Network 2: ???

```

A    I    0.1
JNB  _002
L    345
L    987
*I
T    MW   100
AN  OV
SAVE
CLR
_002:  A    BR
      =    M   10.1
    
```



Инструкции переходов, зависящие от слова статуса

- JU Label¹⁾ Безусловный переход
- JC Label¹⁾ Переход, если RLO = 1
- JCN Label¹⁾ Переход, если RLO = 0
- JCB Label¹⁾ Переход, если RLO = 1 и запомнить RLO в BR
- JNB Label¹⁾ Переход, если RLO = 0 и запомнить RLO в BR
- JBI Label¹⁾ Переход, если BR = 1
- JNBI Label¹⁾ Переход, если BR = 0
- JO Label¹⁾ Переход, если OV = 1
- JOS Label¹⁾ Переход, если OS = 1

1) Метка может состоять макс. из 4 символов: букв и цифр. Первый - буква или _

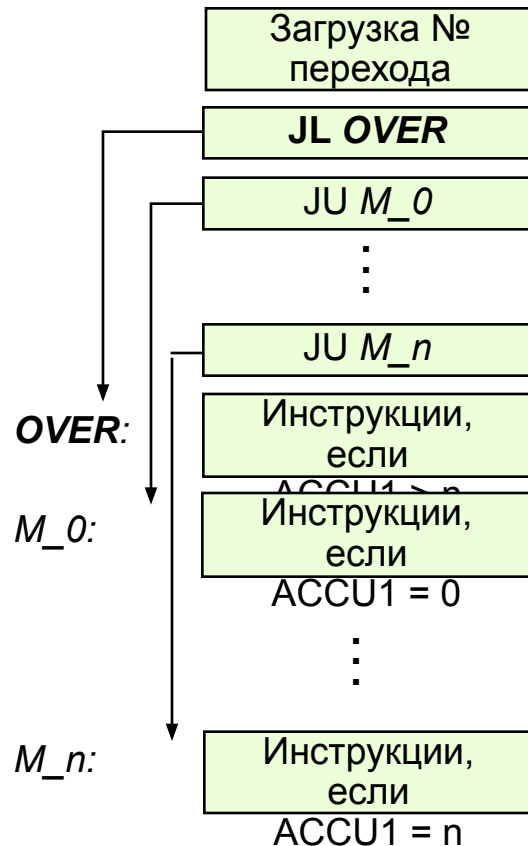
Функции переходов, зависящие от кодов состояния

- JZ Label¹⁾ Переход, если CC 1=0 и CC 0=0
(Результат = 0)
- JN Label¹⁾ Переход, если CC1 не равен CC0
(Результат <> 0)
- JP Label¹⁾ Переход, если CC 1=1 и CC 0=0
(Result > 0)
- JM Label¹⁾ Переход, если CC 1=0 и CC 0=1
(Результат < 0)
- JPZ Label¹⁾ Комбинация переходов JZ и JP
(Результат >= 0)
- JMZ Label¹⁾ Комбинация переходов JM и JZ
(Результат <= 0)
- JUO Label¹⁾ Переход, если: неправильное число типа REAL или деление на 0

1) Метка может состоять макс. из 4 символов: букв и цифр. Первый - буква или _

Программирование распределенных переходов

Режимы операции Пример: Распределенный переход с 3 входами



```

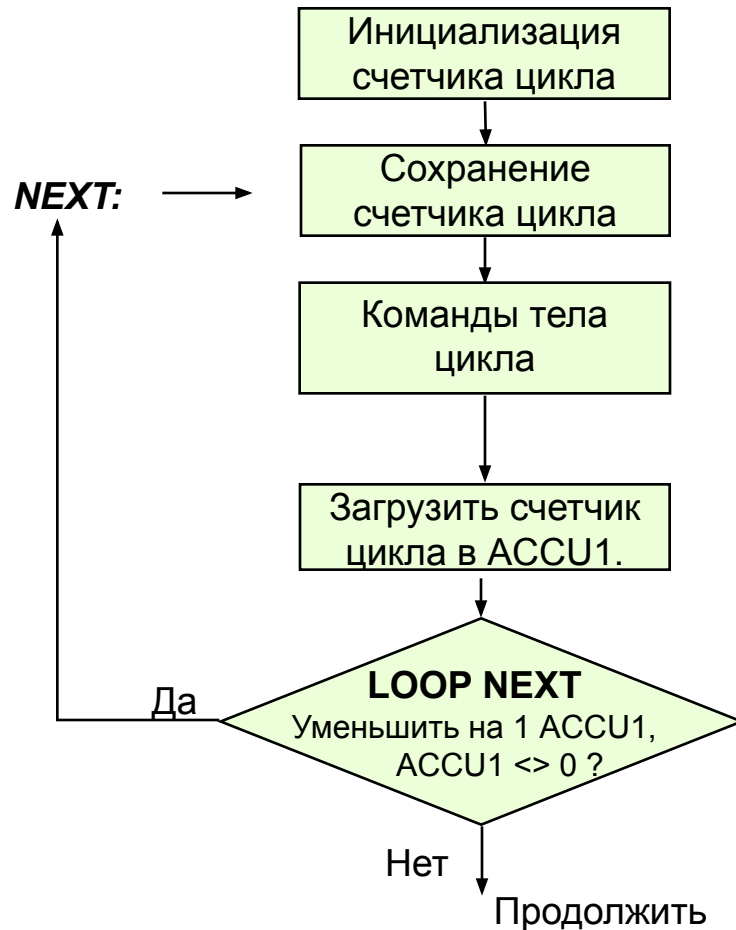
L Recipe_no    // Загрузка номера рецепта
JL OVER       // Распределенный переход с 3 входами
JU Rec0       // Переход на рецепт № 0 (ACCU1 = 0)
JU Rec1       // Переход на рецепт № 1 (ACCU1 = 1)
JU Rec2       // Переход на рецепт № 2 (ACCU1 = 2)
  
```

```

OVER:    JU ERR    // Переход если ACCU1 > 2
...
Rec0:    L DBW4
...
Rec1:    L DBW6
...
Rec2:    L DBW8
...
ERR: CLR
...
  
```

Программирование инструкции цикла

Режим операции



Пример: Инструкция цикла в 4-я повторениями

L +4 // Загрузить счетчик цикла

NEXT: T MW10 // Начало цикла: сохранение

// счетчика

. // Код

. // Код

. // Код


L MW10 // Восстановление счетчика

LOOP NEXT // Уменьшение счетчика на 1,
// переход на метку перехода
// NEXT если Accu 1 > 0

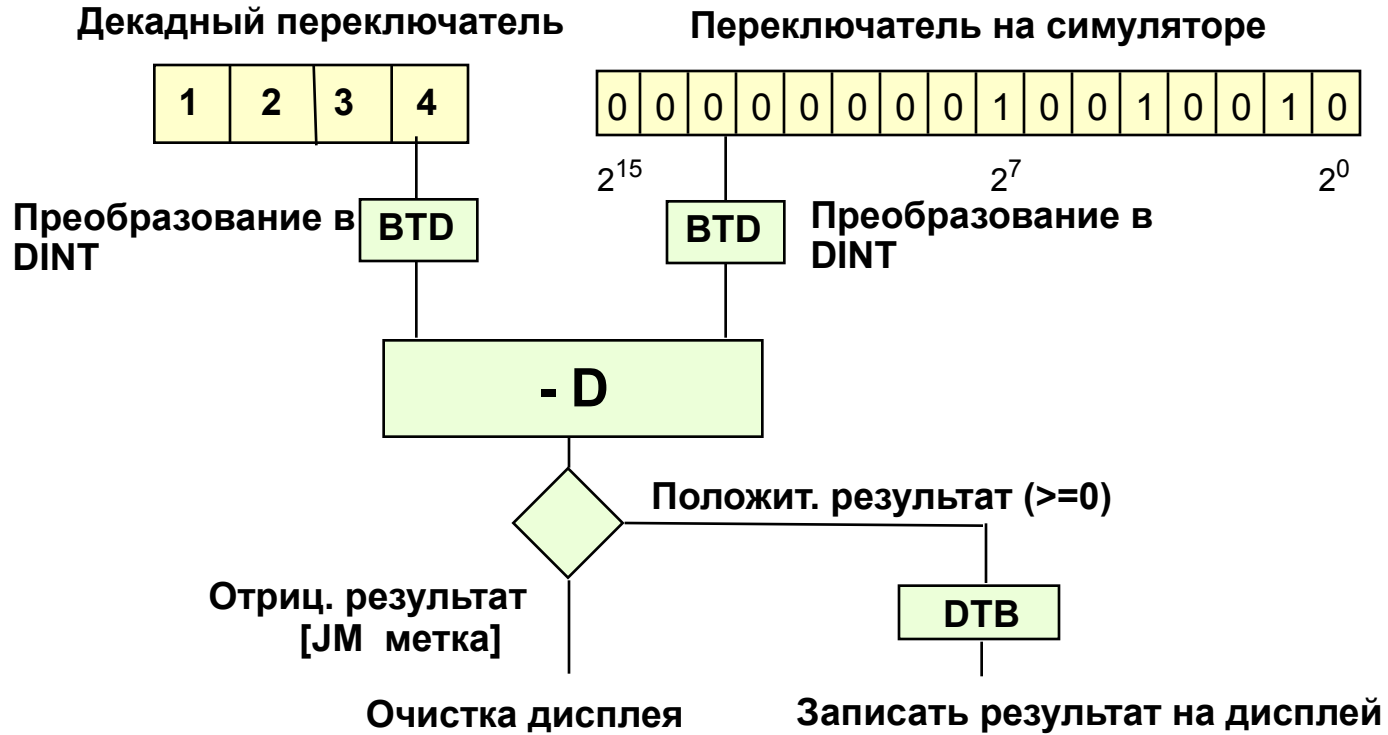
Инструкции окончания блока

- **BE** Конец блока
- **BEU** Безусловное окончание блока (В пределах блока)
- **BEC** Условное окончание блока (зависит от RLO)

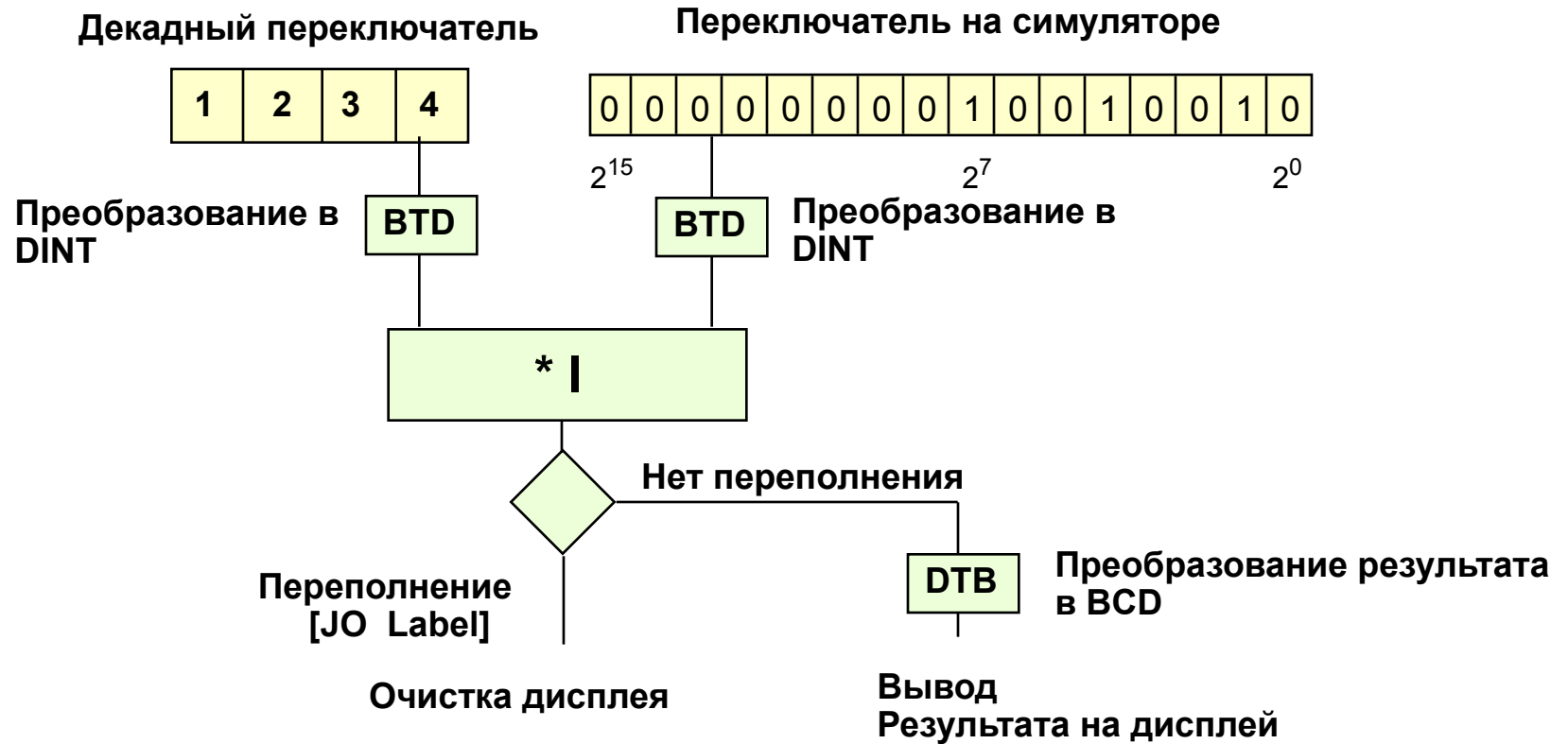
(RET)___ в представлении LAD

— в п  влении FBD

Упражнение 1.1. Переход после вычитания



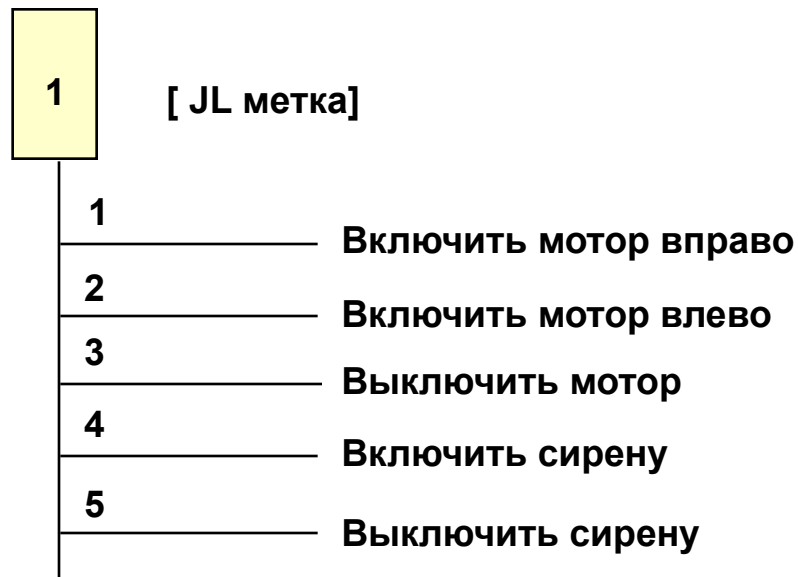
Упражнение 1.2. Переход после умножения



Упражнение 1.3. Программирование распределенного перехода

Функция:

Декадный переключатель



Метка: Переход через список переходов

Адреса	Stand-4	Stand-6	Stand-8
Мотор вправо:	Q125.6	Q5.5	Q5.5
Мотор влево :	Q125.5	Q5.6	Q5.6