

program : sorting (PARAM: cycles, countQuality, countUnQuality)

(Entry) [VAR: cycles, countQuality, countUnQuality, info1, info2, mark1, mark2, mark, picked]

{

```
(1)    TRIGGER (pathSensor == TRUE) {
(2)        cycles := cycles + 1;
(3)        IF (cycles % 2 == 1) {
(4)            PATHSET << 'path1';
(5)        }
(6)        ELSIF (cycles % 2 == 0) {
(7)            PATHSET << 'path2';
(8)        }
(9)    }
(10)   TRIGGER (sensor1 == TRUE) {
(11)       info1 << SREAD1;
(12)       mark1 := checkProduct(info1);
(13)       mark1 >> SWRITE1;
(14)   }
(15)   TRIGGER (sensor2 == TRUE) {
(16)       info2 << SREAD2;
(17)       mark2 := checkProduct(info2);
(18)       mark2 >> SWRITE2;
(19)   }
```

□

```
(15)   TRIGGER (sortSensor == TRUE) {
(16)       mark << SCANNER;
(17)       IF (mark == TRUE) {
(18)           SORTSET << 'Quality';
(19)           countQuality := countQuality + 1;
(20)       }
(21)       ELSE {
(22)           SORTSET << 'UnQuality';
(23)           countUnQuality := countUnQuality + 1;
(24)       }
(25)       picked := TRUE;
(26)   }
(27)   TRIGGER (picked == TRUE) {
(28)       IF(countUnQuality / countQuality > 0.01) {
(29)           STOP;
(30)       }
(31)       picked := FALSE;
(32)   }
```

□

}