```
                    ┌─────────────┐
                    │    Start    │
                    └──────┬──────┘
                           │
         No                ▼
      ┌──────────◇ Check Serial.aVailable()? ◇◄──────────────────────────┐
      │           └──────────────┬───────────┘                           │
      │                          │ Yes                                   │
      │                          ▼                                        │
      │              ◇ incomingByte == ,f' ◇──── Yes ──► ┌──────────────┐ │
      │               └──────────┬─────────┘             │   Forward    │ │
      │                          │ No                    │  Procedure   │ │
      │                          ▼                       └──────┬───────┘ │
      │              ◇ incomingByte == ,b' ◇──── Yes ──► ┌──────────────┐ │
      │               └──────────┬─────────┘             │   Backward   │ │
      │                          │ No                    │  Procedure   │─┼─► ┌──────────┐
      │                          ▼                       └──────────────┘ │   │ StopMove │
      │              ◇ incomingByte == ,l' ◇─ No ─ Yes ─► ┌──────────────┐│   └──────────┘
      │               └──────────┬─────────┘             │ Left Procedure│
      │                          │ No                    └──────────────┘
      │                          ▼
      │              ◇ incomingByte == ,r' ◇──── Yes ──► ┌──────────────┐
      │               └──────────┬─────────┘             │Right Procedure│
      │                          │                       └──────────────┘
      └──────────────────────────┘
```

Start

Check Serial.aVailable()?

incomingByte == ,f'

incomingByte == ,b'

incomingByte == ,l'

incomingByte == ,r'

Forward Procedure

Backward Procedure

Left Procedure

Right Procedure

StopMove

No

Yes

Yes

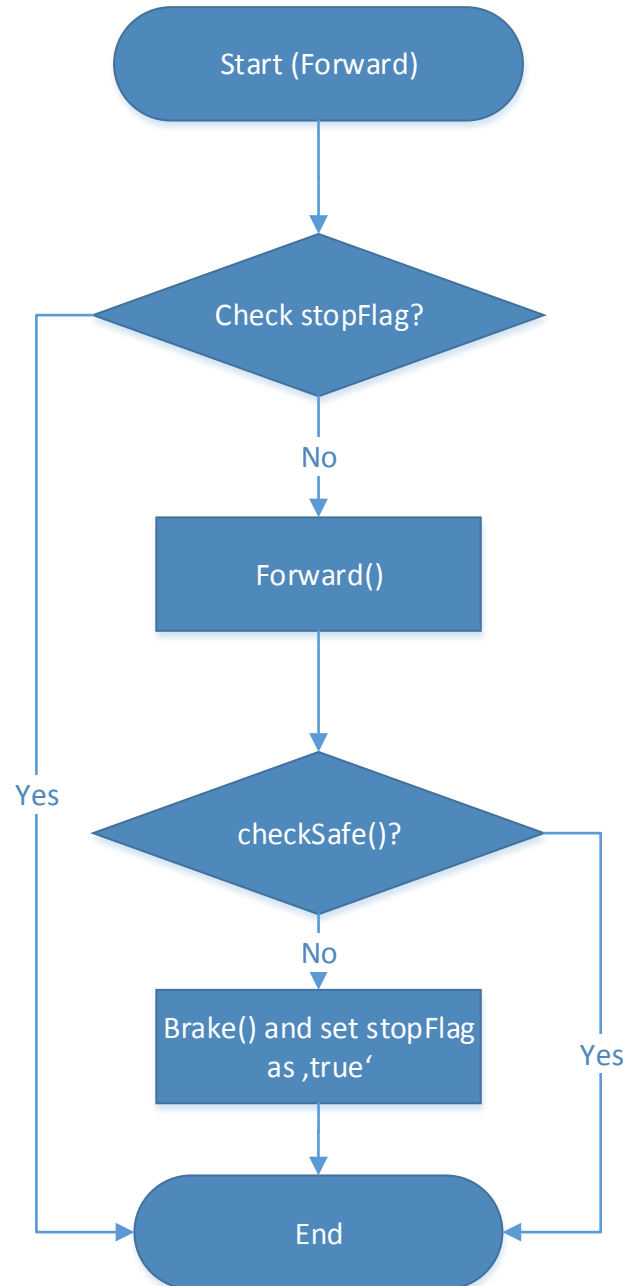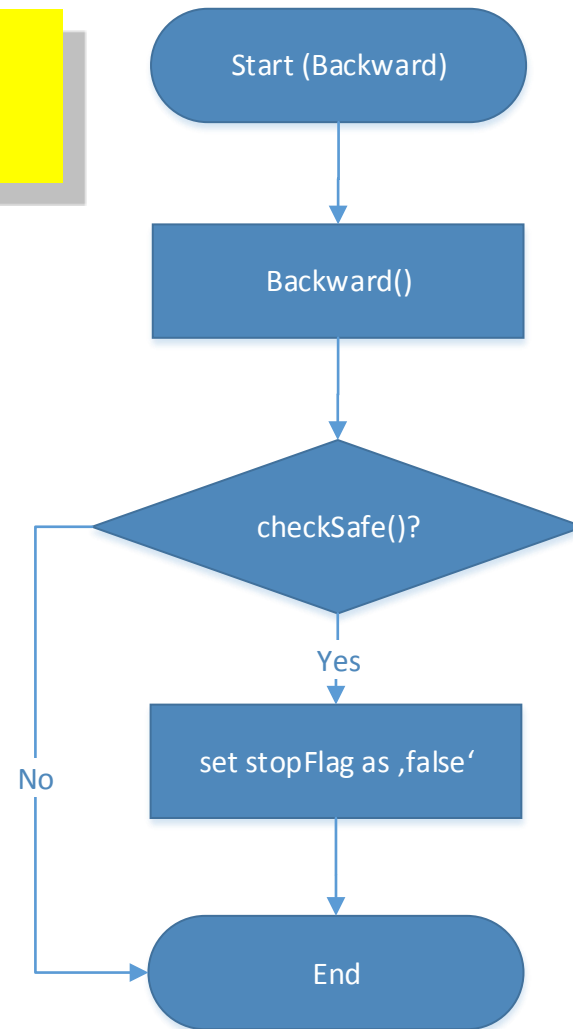No

Yes

No

No

Yes

No

Yes

**Forward Procedure**
**1) stopFlag: global variable, true: Obstacle,**
**2) false: Safe**
**3) Safe Distance (Foward): 20cm**
**4) Safe Distance (Left and Right): 10cm**

```
            Start (Forward)
                  |
                  v
            Check stopFlag?  ──── Yes ──┐
                  |                     │
                  No                    │
                  v                     │
              Forward()                 │
                  |                     │
                  v                     │
             checkSafe()?  ──── Yes ──┐ │
                  |                   │ │
                  No                  │ │
                  v                   │ │
        Brake() and set stopFlag      │ │
              as „true'               │ │
                  |                   │ │
                  v                   │ │
                 End  <───────────────┘─┘
```

**Backward Procedure**
**1) stopFlag: global variable, true:**
**Obstacle, false: Safe**

Start (Backward)

Backward()

checkSafe()?

No

Yes

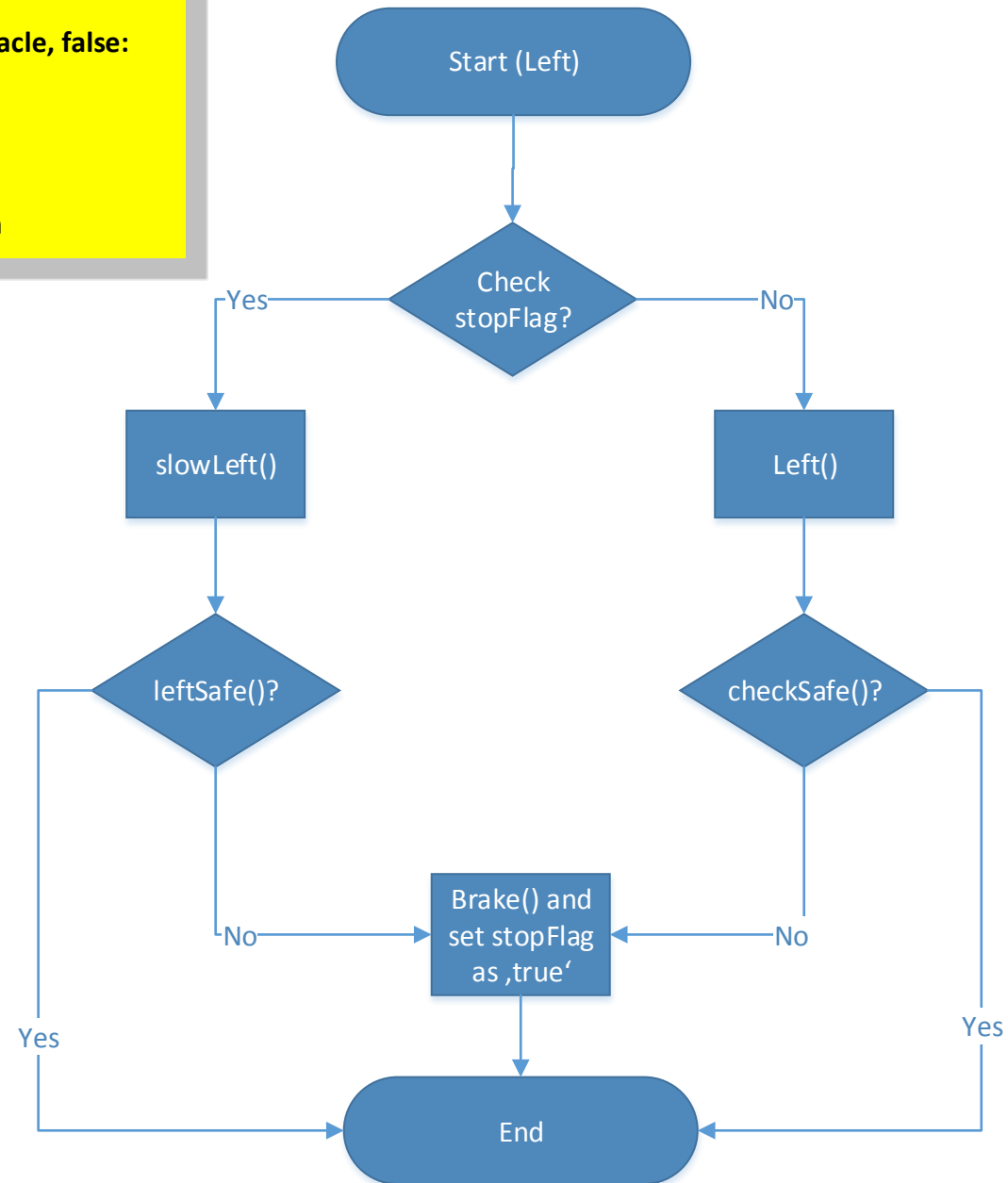set stopFlag as ‚false'

End

Left Procedure
1) stopFlag: global variable, true: Obstacle, false: Safe
2) Left(): PWM 255
3) slowLeft(): PWM 128
4) Safe Distance (Foward): 20cm
5) Safe Distance (Left and Right): 10cm

Start (Left)

Check stopFlag?

Yes → slowLeft()

No → Left()

leftSafe()?

checkSafe()?

Brake() and set stopFlag as ‚true'

No

No

Yes

Yes

End

**Right Procedure**
1) stopFlag: global variable, true: Obstacle, false: Safe
2) Right(): PWM 255
3) slowRight(): PWM 128
4) Safe Distance (Foward): 20cm
5) Safe Distance (Left and Right): 10cm

Start (Right)

Check stopFlag?

Yes → slowRight()

No → Right()

rightSafe()?

checkSafe()?

Brake() and set stopFlag as ‚true'

End

**Forward**

Chasis

Left Motor

Right Motor

**Backward**

Chasis

Left Motor

Right Motor

**Turn Left**

Chasis

Left Motor

Right Motor

**Turn Right**

Chasis

Left Motor

Right Motor