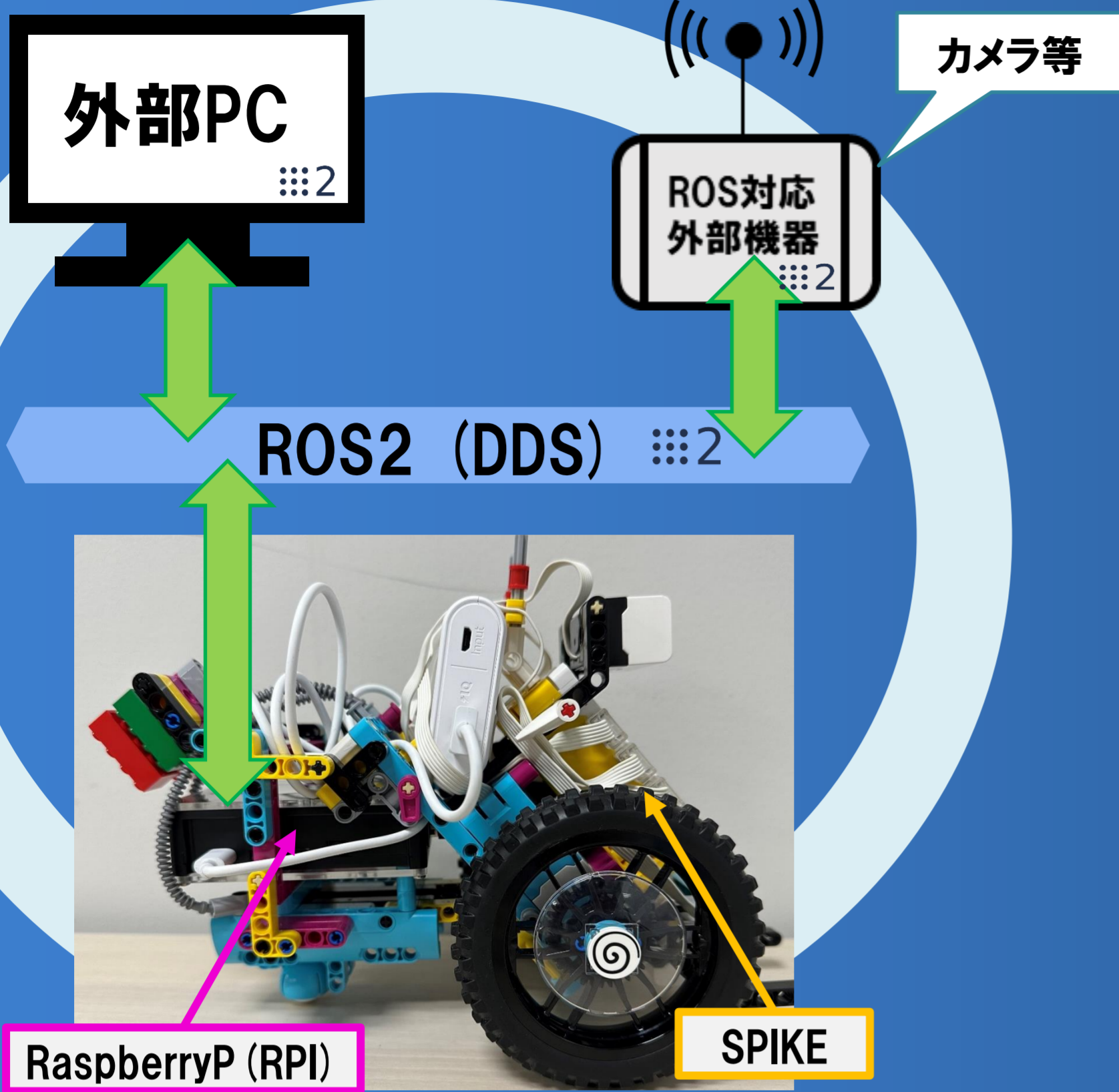
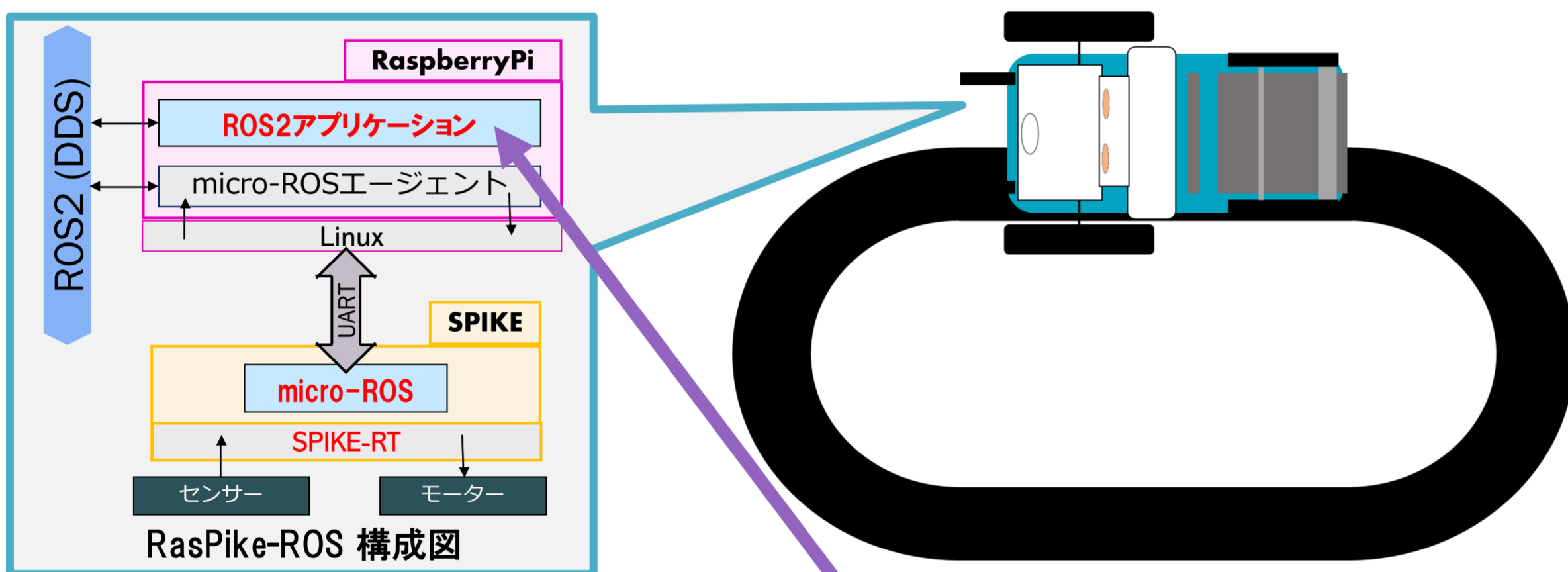


# RasPike-ROS



## RasPike-ROSの特徴

- SPIKE-RTを使用
- SPIKE-RT上でmicro-ROSを稼働
- SPIKE↔RPI間にROS通信を採用
- ユーザはRPIでROS2アプリを開発
  - アプリ開発専用APIも用意
- ROS通信による外部機器との接続が可能
- ROS2教材としての活用も可能



## RasPike-ROS デモ実行

- PID制御による走行体ライントレースプログラム
- アプリ開発専用のAPIを使用したプログラムを実行中

実行中のプログラム



app\_node.py

## Python

```
def steering_amount_calculation(self):
    target_brightness = (white_brightness + brack_brightness) / 2
    diff_brightness = target_brightness - self.rgb_val[2]
    self.diff[1] = self.diff[0]
    self.diff[0] = int(diff_brightness)
    p_val = diff_brightness
    i_val = self.pre_i_val + (self.diff[0] + self.diff[1]) * delta_t / 2
    d_val = int((self.diff[0] - self.diff[1]) / delta_t)

    self.steering_amount = (kp * p_val) + (kd * d_val) + (ki * i_val)

def motor_drive_control(self):
    motor.set_left_motor_speed(int(bace_speed + (self.steering_amount * left_edge)))
    motor.set_right_motor_speed(int(bace_speed - (self.steering_amount * left_edge)))
```

専用API

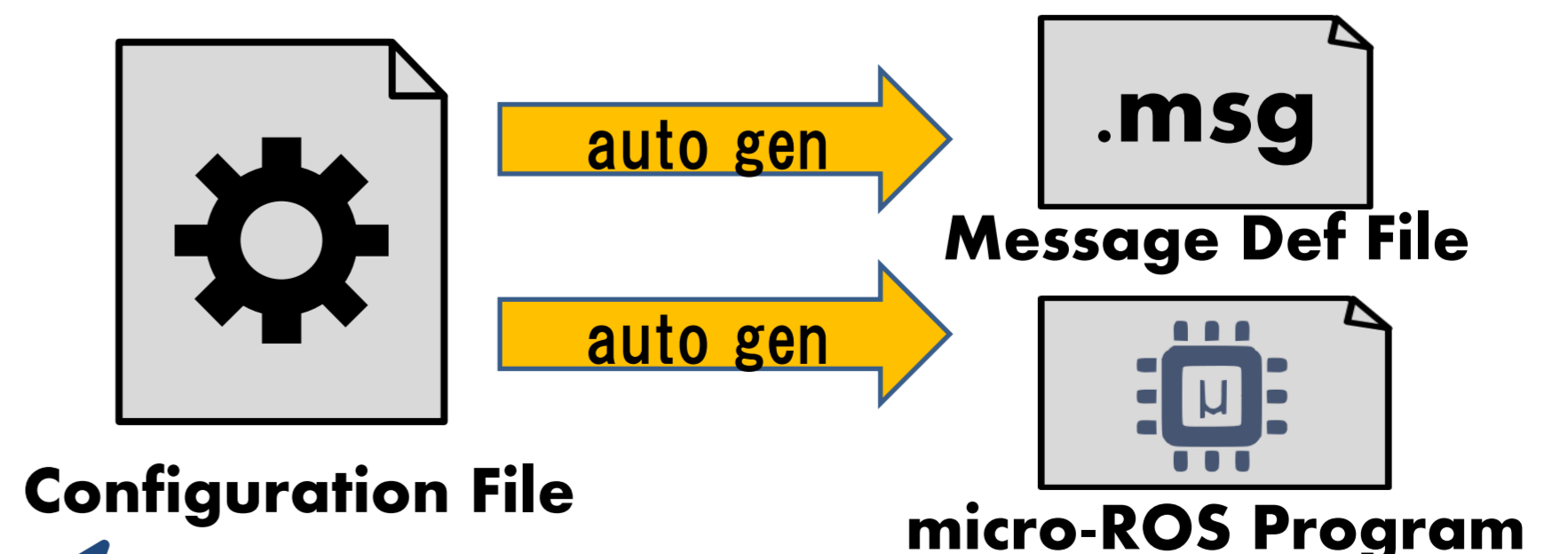
## ◆micro-ROSファームウェアの自動生成の実現

### SPIKEの構成を変更すると

- センサの接続ポートを変更
- 構成に合わせてmicro-ROSプログラムを編集
- 必要なメッセージ型定義ファイルを作成
- ROS2アプリ開発

手間がかかる

### 自動生成ツールを使用



ROS2教材としての活用が可能

