



# ***DENSO***

Crafting the Core

## デンソークリエイト 1DAYインターンシップ° 設計書の読み方ガイド

CONFIDENTIAL  
関係者外秘

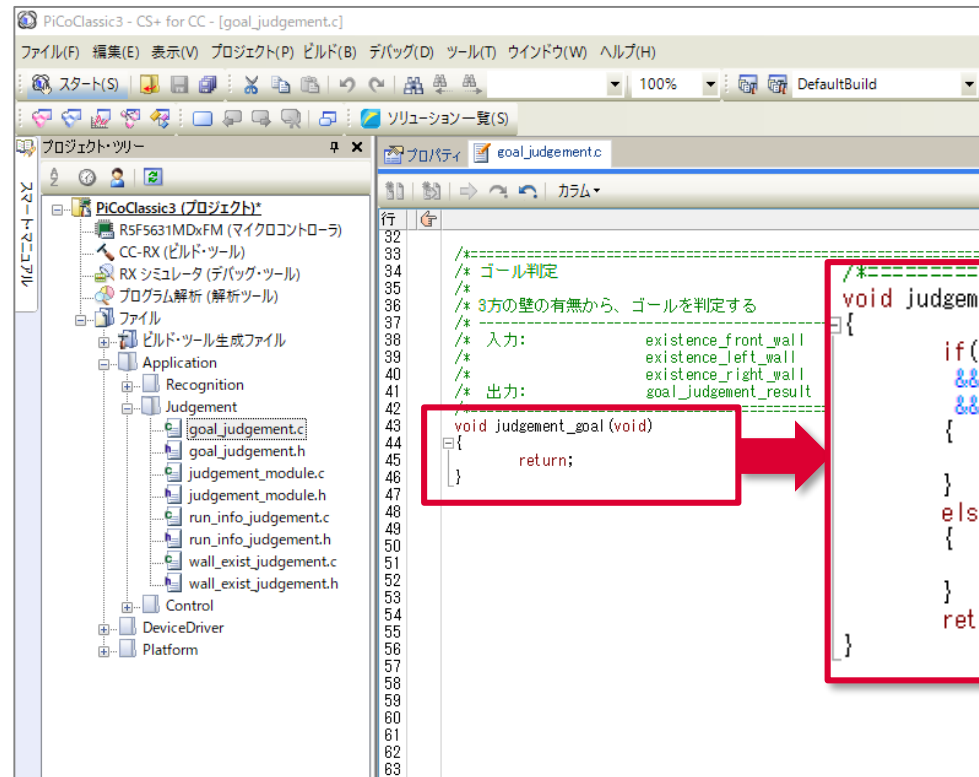
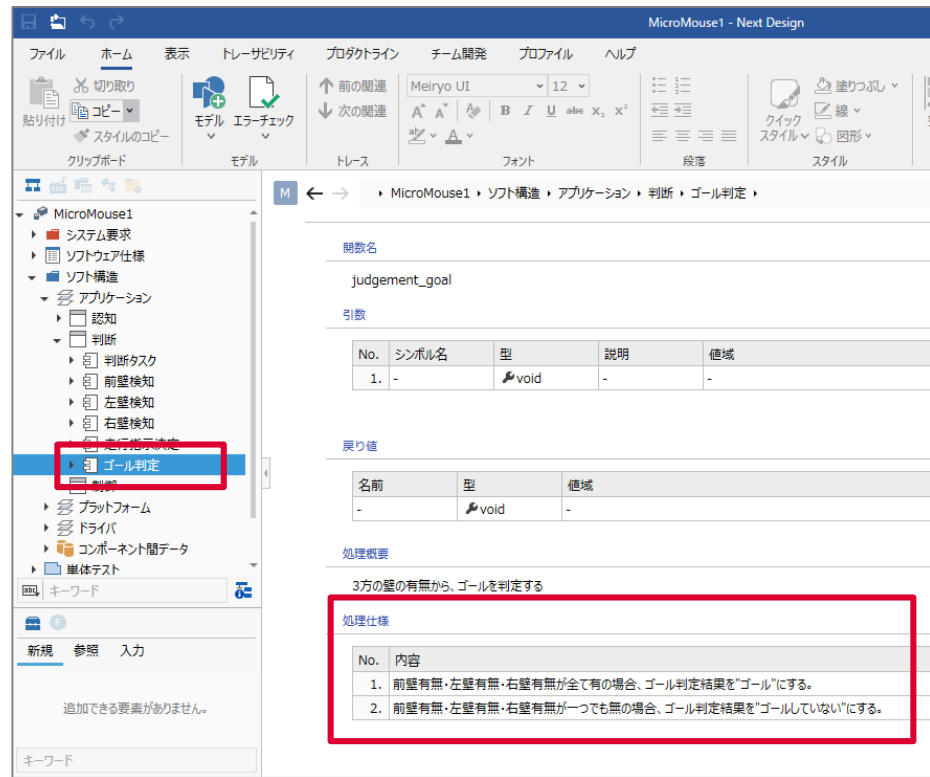
1. 修正すべき箇所
2. 設計書とプログラムの全体構成・対応
3. 設計書内のゴールで停止する記載を探す
4. プログラムを確認する

# 1. 修正すべき箇所

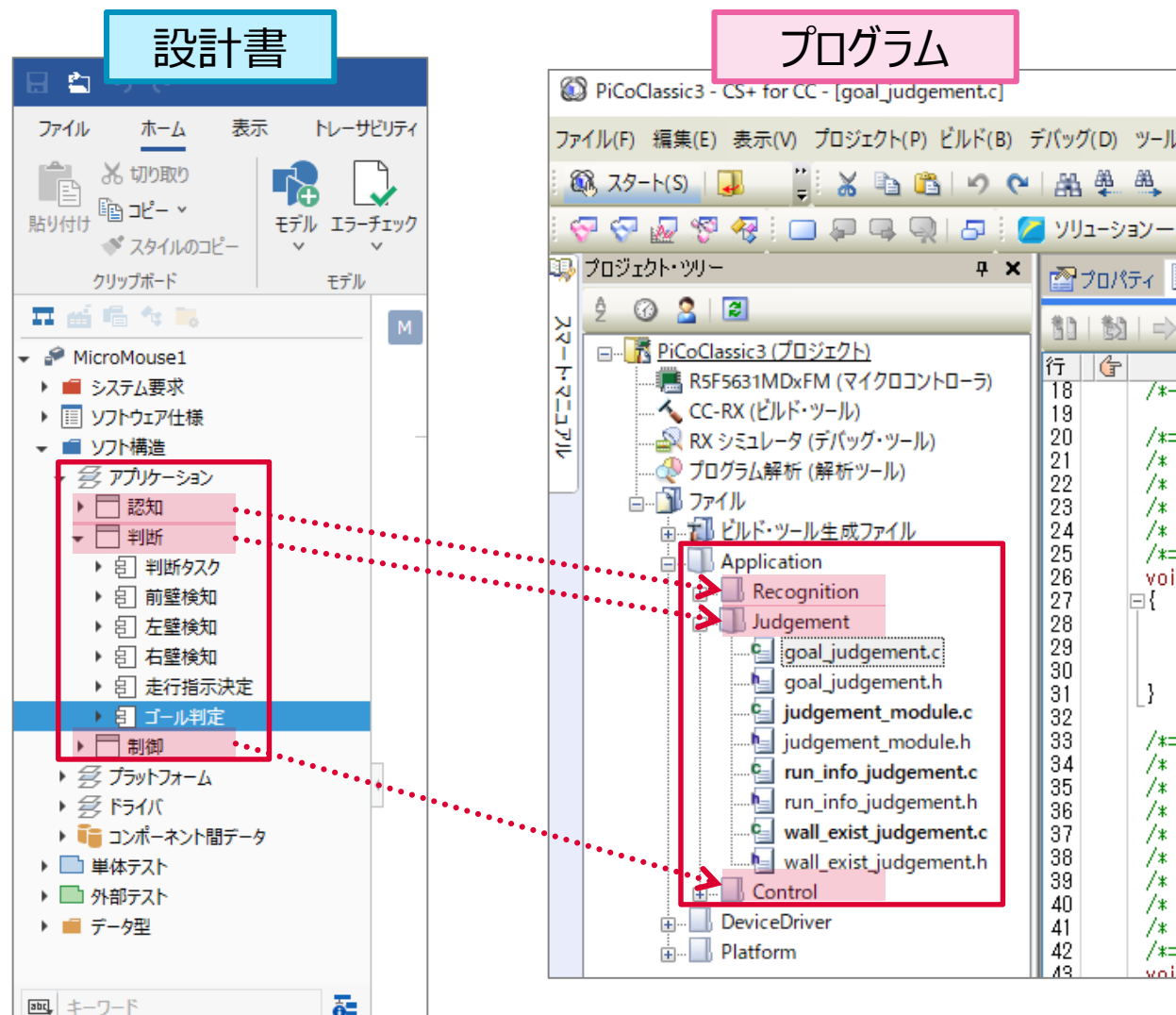
修正すべき箇所（設計書の通りになっていないところ）は以下になります。  
ここにたどり着くまでの流れを、本資料で示します。

設計書の参照箇所

→ ソースコードの修正箇所



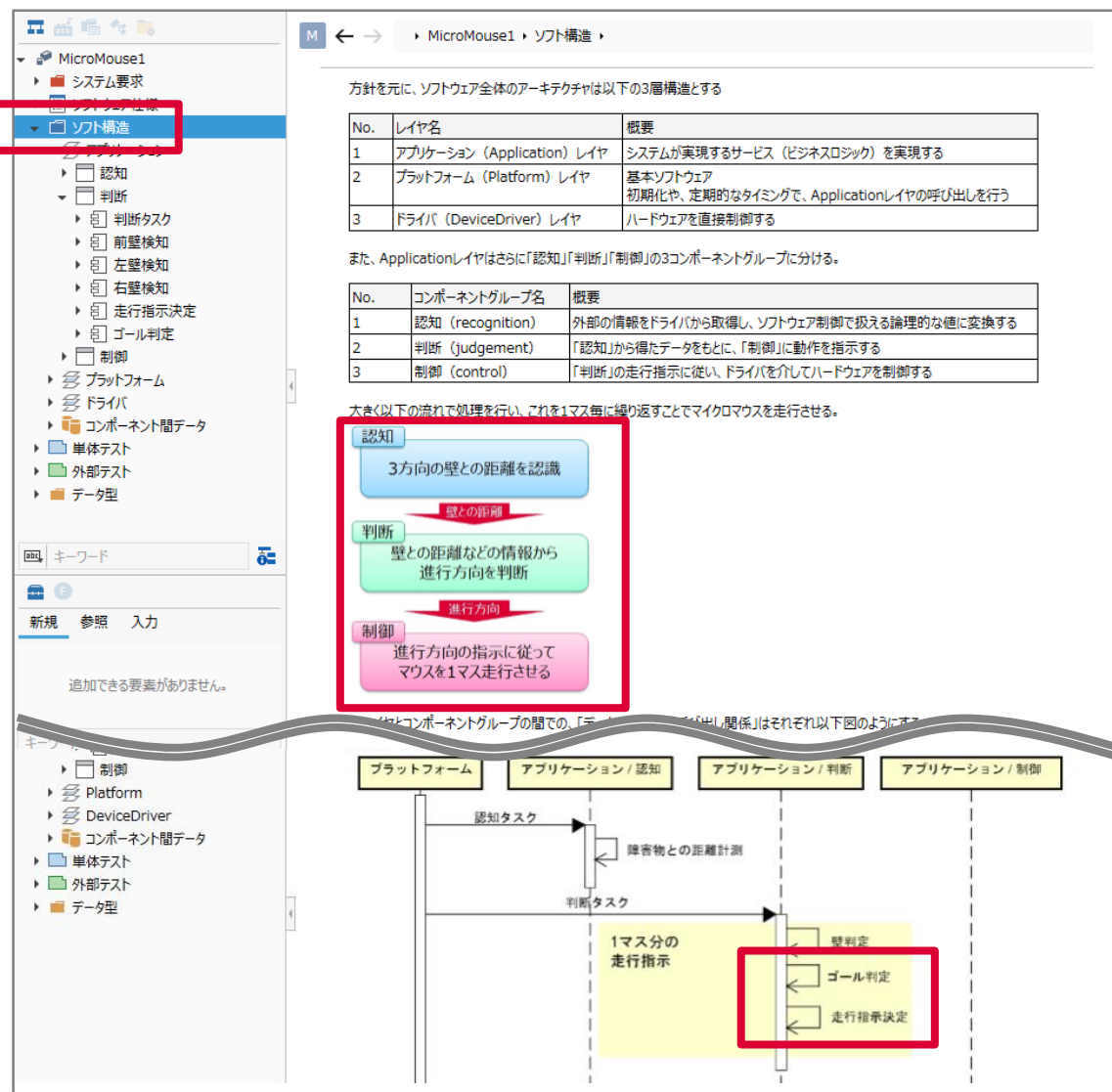
## 2. 設計書とプログラムの構成と対応関係を知る



設計書とプログラムは、同じ構造になっており、それぞれ対応づいています。

設計書左側の、「ソフト構造」以下のツリーと、プログラムのツリーが対応しています。

# 3. マイクロマウスの動作の流れを知る

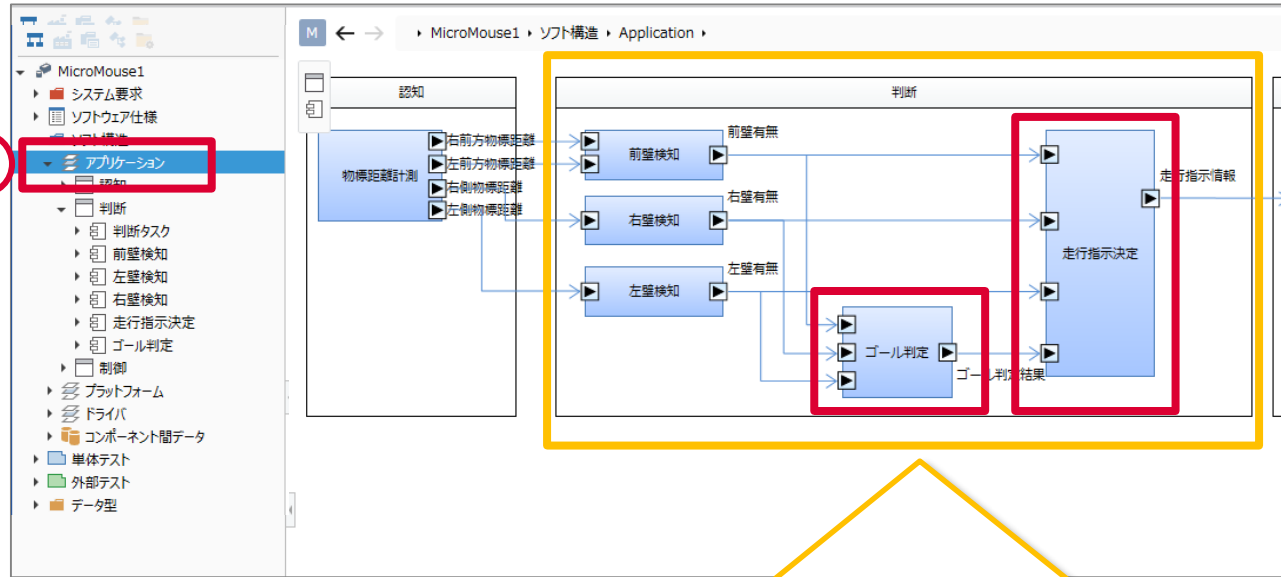


まず「**ソフト構造**」をクリックし全体像を把握します。

ソフト構造を読んでいくと、「アプリケーション」の中でさらに細かく分けて、「認知」→「判断」→「制御」の順に処理していることや、「判断」の中に「ゴール判定」があり、そのあと進行方向を判断していることが分かります。

次は「アプリケーション」の「判断」を詳しく見ます。

### 3. ゴールで停止する機能のあたりをつける



① 「アプリケーション」をクリックし、データの流れます。

「ゴール判定」から「走行指示決定」に  
“ゴール判定結果”というデータが流れていますので、  
次はこの2つに着目します。

② 「判断」をクリックし、各コンポーネントの概要を見ます。

「ゴール判定」で“ゴール”かどうかを判定し、  
それを元に最終的に「走行指示決定」で  
動作を決定していることが分かります。

次は「ゴール判定」と「走行指示決定」を詳しく見ます。

② 「判断」をクリックし、各コンポーネントの概要を見ます。

The screenshot shows the '判断' (Judgment) component group selected in the left sidebar. The main window displays a table of components and their descriptions. The '走行指示決定' (Movement Instruction Decision) and 'ゴール判定' (Goal Judgment) components are highlighted with a red box.

| No. | 名前     | 説明  | コンポーネント名             |
|-----|--------|---|----------------------|
| 1.  | 判断タスク  | スケジュールからの呼び出しに対し、適宜判断グループ内のコンポーネントの処理を呼び出す。 | judgement_module     |
| 2.  | 前壁検知   | 左右の前方物標距離から、前壁の有無を判断する                      | wall_exist_judgement |
| 3.  | 左壁検知   | 左側物標距離から、左壁の有無を判断する                         | wall_exist_judgement |
| 4.  | 右壁検知   | 右側物標距離から、右壁の有無を判断する                         | wall_exist_judgement |
| 5.  | 走行指示決定 | 進行方向を決定する                                   | run_info_judgement   |
| 6.  | ゴール判定  | 3方の壁の有無から、ゴールを判定する                          | goal_judgement       |

### 3. ゴールで停止するロジックを確認する

**ゴール判定**

説明  
3 方の壁の有無から、ゴールを判定する

コンポーネント名  
goal\_judgement

入力データ

| No. | データ  | シンボル名                |
|-----|------|----------------------|
| 1.  | 前壁有無 | existence_front_wall |
| 2.  | 左壁有無 | existence_left_wall  |
| 3.  | 右壁有無 | existence_right_wall |

処理概要  
3 方の壁の有無から、ゴールを判定する

処理仕様

| No. | 内容   |
|-----|--|
| 1.  | 前壁有無・左壁有無・右壁有無が全て有の場合、ゴール判定結果を"ゴール"にする。        |
| 2.  | 前壁有無・左壁有無・右壁有無が一つでも無の場合、ゴール判定結果を"ゴールしていない"にする。 |

前・左・右の3方向に壁があれば  
ゴール判定結果を"ゴール"にします。

**走行指示決定**

コンポーネント 走行指示決定

説明  
進行方向を決定する

コンポーネント名  
run\_info\_judgement

入力データ

| No. | データ  | シンボル名                |
|-----|------|----------------------|
| 1.  | 前壁有無 | existence_front_wall |
| 2.  | 左壁有無 | existence_left_wall  |
| 3.  | 右壁有無 | existence_right_wall |

処理仕様

| No. | 内容   |
|-----|--|
| 1.  | 起動後初回は、走行指示情報を"0.5区画前進"に設定して終了する   |
| 2.  | 起動後初回でなく、ゴール判定結果が"ゴール"の場合、走行指示情報を"停止"に設定して終了する   |
| 3.  | 起動後初回でなく、ゴール判定結果が"ゴール"でない場合、以下の通り走行指示情報を設定する<br>左手法の考え方により、以下の条件のうち最初に当てはまる条件に従って走行指示情報を設定する<br>・左壁有無が無ならば、走行指示情報を"左折"に設定する<br>・前壁有無が無ならば、走行指示情報を"直進"に設定する<br>・右壁有無が無ならば、走行指示情報を"右折"に設定する<br>当てはまらない場合は、走行指示情報を"停止"に設定する |

ゴール判定結果が"ゴール"の場合、  
走行停止します。

ここまで来ればあともう一息。

次は設計書通りにプログラムが書けているか確認します。



## 4. プログラムを確認する

CONFIDENTIAL  
関係者外秘

### 設計書：ゴール判定

|          |   |
|----------|---|
| 判断       | 処理概要  |
| 判断タスク    | 3方の壁の有無から、ゴールを判定する                                |
| 前壁検知     | 処理仕様  |
| 左壁検知     | No. 内容  |
| 右壁検知     | 1. 前壁有無・左壁有無・右壁有無が全て有の場合、ゴール判定結果を"ゴール"にする。        |
| 走行指示決定   | 2. 前壁有無・左壁有無・右壁有無が一つでも無の場合、ゴール判定結果を"ゴールしていない"にする。 |
| ゴール判定    |   |
| 制御       |   |
| Platform |   |

### プログラム：ゴール判定

```
/*
 * 34  /* ゴール判定
 * 35  /*
 * 36  /* 3方の壁の有無から、ゴールを判定する
 * 37  /*
 * 38  /* 入力:      existence_front_wall  前壁有無
 * 39  /*           existence_left_wall   左壁有無
 * 40  /*           existence_right_wall  右壁有無
 * 41  /* 出力:      goal_judgement_result ゴール判定結果
 * 42  /*
 * 43  void judgement_goal(void)
 * 44  {
 * 45      return;
 * 46  }
 * 47
 * 48
```

NG

関数の中身が空っぽなので、**ここが修正箇所**です。

設計書と同じになるようプログラミングします

### 設計書：走行指示決定

|              |  |
|--------------|--|
| 判断           | 進行方向を決定する  |
| 判断タスク        | 処理仕様   |
| 前壁検知         | No. 内容   |
| 左壁検知         | 1. 起動後初回は、走行指示情報を"0.5マス前進"に設定して終了する                              |
| 右壁検知         | 2. 起動後初回でなく、ゴール判定結果が"ゴール"の場合、走行指示情報を"停止"に設定して終了する                |
| 走行指示決定       | 3. 起動後初回でなく、ゴール判定結果が"ゴール"でない場合、以下の条件のうち最初に当てはまる条件に従って走行指示情報を設定する |
| ゴール判定        | ・左手法の考え方により、以下の条件のうち最初に当てはまる条件に従って走行指示情報を設定する                    |
| 制御           | ・左壁有無が無ならば、走行指示情報を"左折"に設定する                                      |
| Platform     | ・前壁有無が無ならば、走行指示情報を"直進"に設定する                                      |
| DeviceDriver |  |

### プログラム：走行指示決定

```
/*
 * 46  /* 走行指示情報決定
 * 47  /*
 * 48  /* 進行方向を決定する
 * 49  /*
 * 50  /* 入力:      -
 * 51  /* 出力:      run_info      走行指示情報
 * 52  /*
 * 53  void judgement_run_info(void)
 * 54  {
 * 55      // 起動直後0.5マス進む
 * 56      if(run_state == RUN_STATE_INIT)
 * 57      {
 * 58          run_info = Half;
 * 59          run_state = RUN_STATE_STEADY;
 * 60          return;
 * 61      }
 * 62      // ゴール到達時停止
 * 63      if(goal_judgement_result == GOAL)
 * 64      {
 * 65          run_info = Stop;
 * 66          return;
 * 67      }
 * 68      // 左手法制御
 * 69
 * 70
 * 71
```

OK

設計書通りに、ゴールの場合に停止できています



***DENSO***

Crafting the Core