

BrickOS によるタスクシステムの設計と 通信関係タスクの実装

064 森下 篤

1. 概要

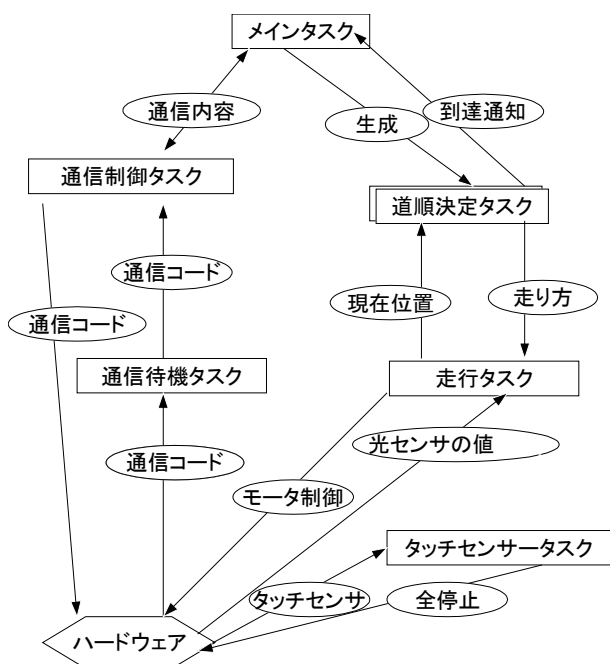
LEGO Mindstorms の RCX (Robotics Command System) を用いて、組込みソフトウェア開発を行った。内容は、分析された仕様に基づくタスクシステムの設計と、それを満たすプログラミング手法の探求である。

2. BrickOS によるタスク設計

BrickOS とは、C 言語をクロスコンパイルすることにより、RCX でユーザプログラムを動作させる OS である。優先度を用いたマルチタスクを備えている。

今回は、『宝探し』という動作検証案を設定した。2 台の RCX を使い、一方を車、他方を宝とし、車が宝と通信することによって、宝を探し出すとゴールである。

この動作検証案を満たすタスクを考え出し、以下の図のようになった。



まず、目的を達成させるメインタスク、通信系タスク、移動系タスクに分けた。各タスクを紹介する。移動系については、道順決定タスクと走行タスク

クに分けた。道順決定タスクは、どこに行くべきか道順を考えるが、これは外部イベントによる即時動作が必要ではない。走行タスクは、光センサーの値を用いてライン上をぶれずに、道順決定タスクから道順を受けて走行する。これは光センサーの値を常に監視する必要があるので、優先度が高めになっている。

通信系は、通信制御タスクと通信待機タスクに分けた。通信待機タスクは、外部イベントである受信に対し即座に対応するため、受信を担う通信待機タスクは優先度を上げられている。実際の通信のデコード、送信は通信制御タスクが行う。

3. 通信関係タスクの設計

通信は赤外線を用いるため、確実ではない。

通信待機タスクは、外部イベントに呼応するため、即時動作を要求される。かつ、処理内容は高速にするため、受信後のコードをキューに入れるのみである。

通信制御タスクは、受信処理と送信処理に分けた。受信処理は、キューから受信文を一つ取り出し、デコードする。送信処理は、受信処理の結果に合わせて、送信分を作成して、送信する。

複数回の通信を行うため、宝は通信要求のコードを送信し、車はそれを受信すると、送信許可のコードを送信する。これにより、通信の成立とする。位置情報のコードを受信すると、受信完了のコードを送信する。これにより、次の情報を送るよう指示できる。また通信の失敗も検知できる。

4. 問題点と課題

通信には成功したが、非常に時間がかかった。通信を移動系に反映させる必要があった。

現在は一对一の通信しか行えない。多対多の通信を行うには、また別のアイディアが必要である。