ミニ四駆AIチュートリアルセッション

〜 ミニ四駆AI チュートリアル： ミニ四駆 IoT で 簡単開発 〜  
講師： 矢野 良和 （愛知工業大学）

概要

ミニ四駆AI に関するチュートリアルの第3段です。マイコンボードを一新し、WiFi や Bluetooth による通信をサポートしました。また、数MBの内蔵メモリにも走行ログが残せるため、ネット環境がなくても単体走行による実験が可能です。さらに、C言語のみでなく python による開発もサポート。簡単なチェックなら対話型実行環境で試すこともできます。本チュートリアルではハードウェアの簡単な説明とプログラムによる簡単な制御について紹介いたします。

本講座ではミニ四駆本体およびマイコンボードを貸出いたします。プログラム開発に利用する PC をご持参ください。事前にシリアル通信のプログラム(Windows では TeraTerm など)がインストールされていると、当日の準備が少なくなります。

大会期間中、本チュートリアルで紹介する機器を貸出いたします。本チュートリアルの知識でミニ四駆AI大会への参加も可能です。

ハードウェア仕様 ()

　マイコン： ESP32 Espressif 社製マイクロコントローラ

CPU: Xtensa LX6 32bit 160MHz （Dual Core / FreeRTOS搭載)

Memory: 520KiB SRAM

ROM: 4MB flash memory

GPIO: 21ch, ADC: 16ch, SPI: x2, I2C: x2, UART: x3, DAC: x2

WiFi: 802.11 b/g/n

Bluetooth v4.2

　出力：

　　モータドライバ：

　　　FET-Driver On/OFF/BRK : ミニ四駆とは３線で接続

　　LED： 2ch

　入力：

　　スイッチ： 3ch-GPIO, Reset

　　ロータリエンコーダ： 2ch ( 1ch 分を実装済み )

　　電圧モニタ： モータバッテリ、マイコンバッテリ

　I2C接続機器：

　　姿勢センサ( MPU6050 )

　その他：

　　USB-Micro 端子：充電専用

　　シリアルインターフェイス： 1ch（シリアルモニタ用）

　　I2Cインターフェイス： ３ｃｈ（うち1つは姿勢センサ取り付け済み）

　　LiPo バッテリ（取り付け済み：ソケットと直接ハンダづけ）

　　スピーカポート（スピーカ未実装）

　　オンボード姿勢センサ（未実装： ST-LSM6DS3用）

(写真）

開発環境について

ESP32 開発環境

- 公式 SDK + ESP-IDF ( C言語 )

すべての機能を使うなら。

- Arduino IDE ( C言語 )

　一部機能限定されるが、Arduino のプログラム資産を転用可能（一部仕様が違うものもある）

- Micro Python ( Python )

　マイコン用に開発された機能限定版のpython。

C言語ベースの開発環境

　マイコンのフル機能を使うのであれば、公式SDKを使うしかない。実装された FreeRTOS のコントロールなどもでき、デュアルコアをフル活用できる。us単位のタイミングなどを重視した高速なハードウェア処理を得たい場合も公式SDKの利用が適している。

　Arduino ではハードウェアアクセスにブラックボックス部分があるが、厳密なタイミングを求めなければ開発が容易である。ハード制御のAPIを共通化させているため既存の Arduino 資産が使える。演算やメモリ管理などはC言語ベースのため、非常に高速。多くのユーザが Arduino IDE にて開発を行っている。

　これら開発環境（コンパイラなど）はPCにインストールする必要がある。作成したファームウェアは、シリアルケーブルにて転送することで書き換え可能である。（書き込みアダプタなどの専用ハードウェアは不要、ただしモード切替の操作が必要。）

pythonベースの開発環境

　マイコンの高性能化から、スクリプト言語の利用もできるようになってきた。ESP32では python での開発が利用可能で、ファームウェアを一度書き込むだけで利用できる。ROM,RAMの制約からかなり機能制限されているが、小さなアプリケーションを作るには非常に便利に開発ができる。PC版のpython と C言語 と同様のメリットデメリットがある。

　REPL（Read – Evaluate – Print Loop) のインタプリタがマイコン上で動作するので、シリアルケーブルだけでかんたんなハードウェアテストが容易に行える。プログラムの再コンパイルやファーム転送なくパラメータを色々変えた試験が容易であり、プログラムもシンプルなテキストファイルとして管理されることから実行する処理を適宜変更することも容易である。テキストファイルの差し替えでプログラムを更新することができるのも非常に便利である。プログラムの開発にはテキストエディタとシリアル通信プログラムがあればよい。ファイル転送などのツールがpython で製作されているため、python と転送ツールがあると望ましい。( MAC や Linux ではほとんどが標準で入っている。Windowsユーザのみ各種ツールのインストールが必要である。）

　難しいハードウェア制御は Arduino と同様に事前に様々なモジュールが用意されている。python 自体はスクリプト言語で処理は遅いが、モジュール自体はC言語で開発されているためモジュールを利用するアプリケーションであれば高速に動作する。また、非常に高度にはなるが各自でモジュールをC言語にて開発することも可能であるため、高速な専用演算、新しいハードウェアの利用もできる。

セットアップ （Windowsユーザのみ）

・ TeraTerm のインストール

・ シリアルアダプタのドライバインストール

・ python3 のインストール ( miniconda )

セットアップ（すべてのユーザ）

・ ampy のインストール

　pip3 install adafruit-ampy

ampy を使うことで、シリアル通信を利用したファイルの送受信ができるようになる。

REPLを使えば、手作業でファイルを送ることも可能。

ただし本チュートリアルではマイコン上で FTP サーバを動かしているので、

ファイルの送受信は ftp 経由で実行できる。windows であればExplorer を使うことでファイルの送受信ができる。LinuxやMAC では FTPの各種コマンドや busybox というCUIコマンドでファイル送受信ができる。

試してみよう、python ミニ四駆

1. LEDの点灯 と スイッチを読む処理

下記のプログラムでスイッチに連動したLED表示が実現できる。

REPL から順に入力して試すこともできる。またブロックモードにしてコピーペーストして、ブロックモードを解除してもよい。

ESP32 の基本的なハードウェアを利用するときは machine をインポートする。