

TOPPERS 活用アイデア・アプリケーション開発 コンテスト

部門 : がじえるね IoT 部門

作品のタイトル : IoT ドライブレコーダ+OBD2 モニタ

作成者 : 松浦光洋@名古屋電子工作の会

共同作業 : なし

対象者 : 車好き, 運転好き, ドライブレコーダに興味がある人

使用する開発成果物 : TOPPERS/ASP (GR-PEACH)

目的・狙い

交通事故の原因調査やドライバーの安全意識向上に役立つと、ドライブレコーダが注目を集めている。昨年 1 号機を製作 (参考文献 1) したが、GR-PEACH はカメラを 2 台接続可能で JPEG コーデックのハードウェアも内蔵している。そして TOPPERS/ASP も移植された。これはドライブレコーダの自作に丁度良いボードだということで 2 号機を製作した。自作ならではのものとして IoT で車の走行位置をどこからでも知ることが出来る機能も搭載。また OBD2(CAN) に接続して走行中の各種情報をモニタする機能も付けた。

アイデア/アプリケーションの概要

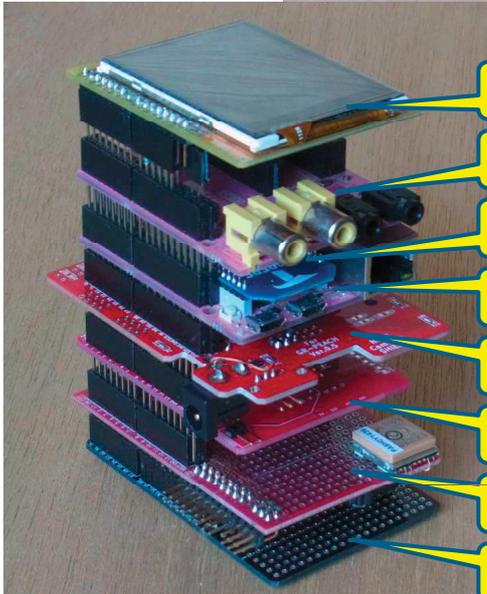
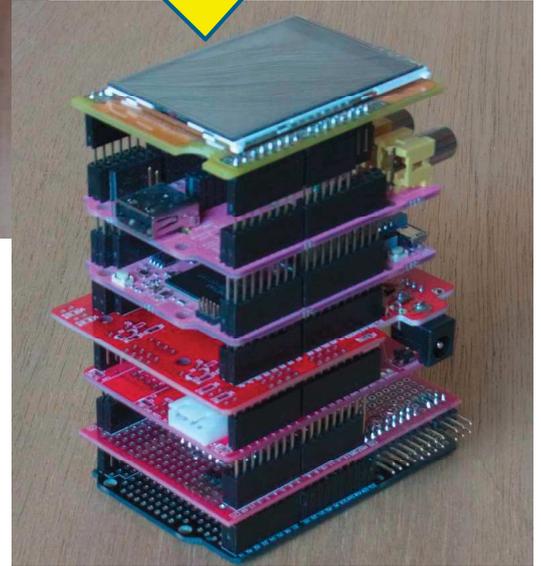
- ① 2 台のカメラで車の前後を録画 (音声の録音は非対応)。
- ② カメラ画像, 各種情報を LCD に表示。スイッチで画面を選択。
- ③ 衛星から受信した nmea データをファイルに記録。
- ④ スマホ等のテザリング経由でクラウドサービスに接続し, 走行中の車の現在位置がどこからでもわかる。待ち合わせの時などに便利。
- ⑤ OBD2 の CAN 端子に流れる車両診断情報を LCD 画面に表示。
- ⑥ 運転者が認識しやすいよう状態表示を音声により行う。

外観



手持ちのArduino形状シールドを積み上げることでソフトの開発をすぐに始める事が出来た

当面はこれで運用しながら専用基板を製作する予定



LCDシールド(※1)

ビデオ入力シールド(※2)

NCES Wi-Fi ボード(※3)

GR-PEACH(※4)

DC/DC電源、音声出力(※5)

NCES CAN シールド改(※3)

衛星測位受信シールド(※5)

単なる土台(※5)

※1 <http://www.aitendo.com/product/7214>

※2

<http://itstore.zaikostore.com/shopdetail/000000001778/ct67/page1/order/>

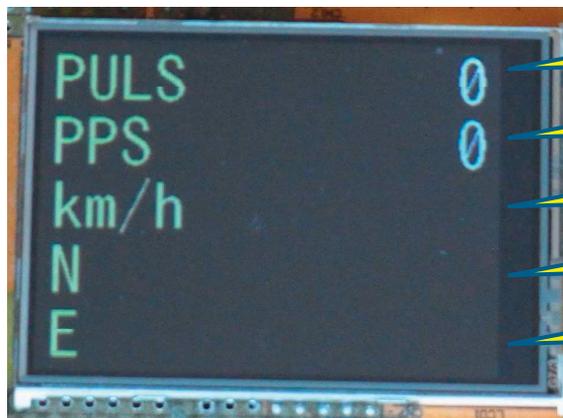
※3 http://ma2.la.coocan.jp/miconboard/nces_shield/

※4

<http://itstore.zaikostore.com/shopdetail/000000001684/>

※5 自作

LCD画面



車速パルスカウント

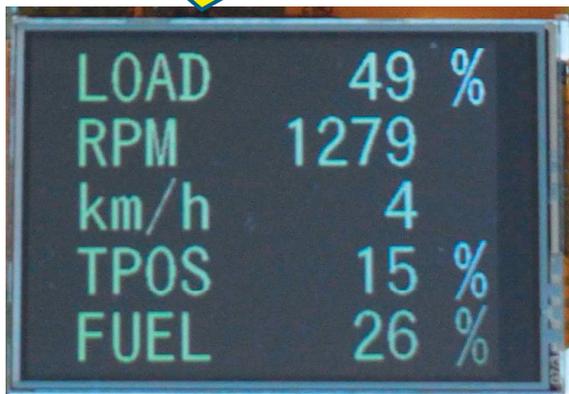
車速パルス/秒

移動速度(衛星測位)

北緯度(衛星測位)

東経度(衛星測位)

OBD2モニタ(1/2)



OBD2モニタ(2/2)



他に前/後カメラのリアルタイム画像を表示する画面

カーネルオブジェクト一覧

タスク	優先度
メイン(※)	1
車速パルス計測	2
カメラ設定(※)	3
時計(※)	4
AVI録画	5
NMEAログ	6
音声WAV/MP3再生	7
LCD表示	8
IoT	9
OBD2モニタ	10

※起動時のみ存在

周期ハンドラ	
AVI録画	250ミリ秒
時間管理	1秒
LCD(DMA設定用)	10ミリ秒
GPIOポーリング	100ミリ秒

割り込みハンドラ	
車速パルス	IRQ7
衛星測位受信機UART	RxD4/TxD4
ESP-WROOM-02用UART	RxD7/TxD7

その他	数
セマフォ	6個
データキュー	8個

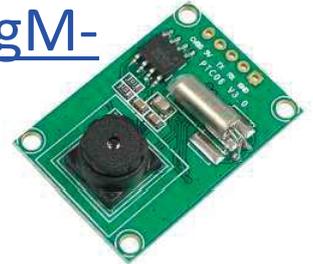
各タスクの概要

- メイン
 - 自動で起動される唯一のタスクであり他のタスクを起動&起床する
- 音声WAV/MP3再生タスク
 - SDカードのWAV/MP3ファイルをPWM出力して自機の状態を音声でアナウンスする
 - 音声WAVファイルは文献2を使用して作成
- AVI録画
 - カメラ2台のJPEG画像を横に並べてAVI動画ファイルを生成する
- カメラ設定
 - カメラを設定する
- 時計
 - 時間管理用周期ハンドラをスタートする
- IoT
 - 10秒間隔で位置情報をUARTに出力
(ESP-WROOM-02がクラウド (IBM Bluemix) に送信)
- LCD表示
 - LCD画面の表示と切替を行う
- NMEAログ
 - 測位衛星から来た全ての情報をnmeaファイルに保存し日時/移動速度/位置座標を抽出
- OBD2モニタ
 - OBD2/CAN信号を受信してLCD画面に表示するPIDのデータを抽出
- 車速パルス計測
 - 車速パルスをカウントして積算値と毎秒値を計測する

録画機能

- RZ-A1HはNTSCビデオ信号入力を2CH内蔵
- 同じくJPEGコーデックのハードウェアも内蔵
- カメラはAdafruit製

<http://akizukidenshi.com/catalog/g/gM-07746/>

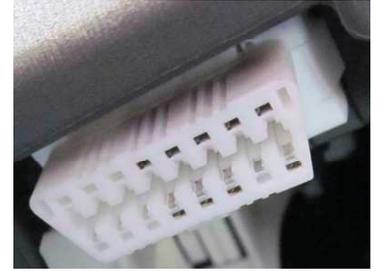


- 動画ファイルはAVIコンテナ形式
- 動画の画面サイズとフレームレートはVGA × 2枚(前後カメラ) × 4FPS

- 8月末の時点では1.3FPS程度しか撮れなかった
 - QVGAにすればデータサイズが小さいため4FPS程度で撮れるが...
 - アナログ (NTSC/PAL) カメラは画質が悪いのでQVGAでは厳しい
 - デジタル (OV系) カメラならQVGAでも画質は良いと思うが2台同時使用が出来ない
- ネットはSDカードへの書き込み速度
 - mbedのSDライブラリは汎用性を持たせるためオーバーヘッドが大きい
 - 素のFatFsに替えてVGA×前後2面で4FPSになった
- もっとフレームレートを上げたいが...
 - SDのSPIクロックは16.66MHzにて動作
 - 33.33MHzに上げたらファイルが正常に作られない

OBD2モニタ機能

- 運転席にあるOBD2コネクタのCANバスに接続
- 下表の情報を抽出(文献3)
- 名称と値をLCD画面に表示



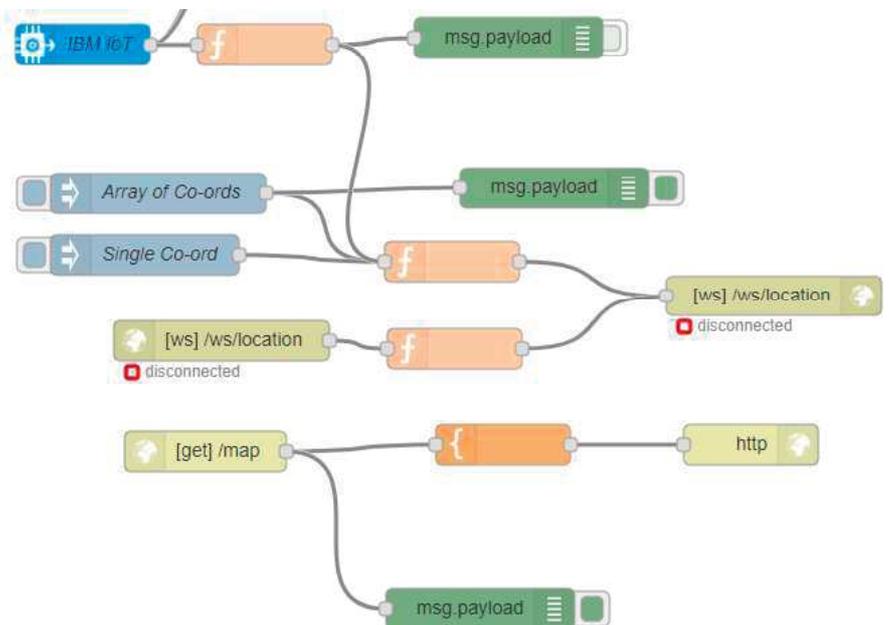
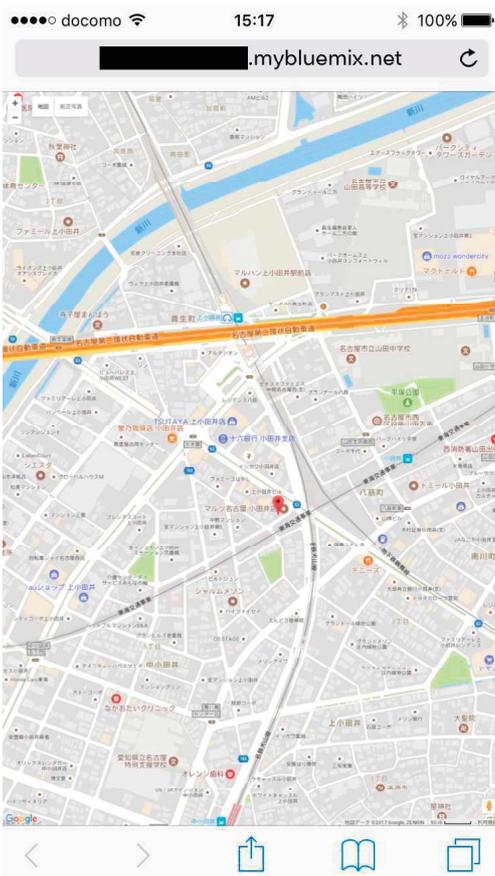
LCDのOBD2画面(1/2)		LCDのOBD2画面(2/2)	
LOAD	Calculated engine load(%)	CLTMP	Engine coolant temperature(° C)
RPM	Engine RPM(rpm)	PRESS	Intake manifold absolute pressure(kPa)
km/h	Vehicle speed(km/h)	TMADV	Timing advance(° before TDC)
TPOS	Throttle position(%)	IATMP	Intake air temperature(° C)
FUEL	Fuel Tank Level Input(%)	ARFLW	MAF air flow rate(grams/sec)

- 最初はmbedのCANライブラリで500kHzの信号を受信できなかった
- CPUの周辺クロック66.666666MHzを分周してCANのクロックとしているため500kHzぴったりにはならない
- ライブラリは誤差2%で設定していた
- 関数を修正して誤差1%にしたら受信できるようになった
- CANの規格では0.5%以下に規定されているので水晶発振器を増設して誤差0%にした

IoT機能

- ESP-WROOM-02 (以下ESP)を単体で「Arduino + 無線LAN」として動作させている
- MQTTプロトコルをESP内に実装(文献4)
- 位置座標をJSON形式の文字列にしてUART経由でESPに渡すとESPがBluemixにパブリッシュする
- Bluemix側ではNode-REDでJavaScriptを動かしWebSocketで接続してきたクライアントに座標の位置をマーキングしたGoogleMapの地図を返す(文献5)

Node-REDのフロー



使用したオープンソースと参考文献

- 動的メモリ管理TLSF: <http://www.gii.upv.es/tlsf/>
- FatFs: http://irtos.sourceforge.net/FAT32_ChaN/doc/00index_j.html
- MP3デコーダ: <http://www.underbit.com/products/mad/>
- AVIファイル構造: <https://www.ffmpeg.org/>
- 文献1:
 - http://www.kumikomi.net/interface/sample/201608/if08_142.pdf
- 文献2: TTS非モノローグ音声合成
 - http://komeisugiura.jp/software/software_jp.html
- 文献3: OBD2のPID
 - https://en.wikipedia.org/wiki/OBD-II_PIDs
- 文献4: ESP-WROOM-02によるBluemixへの接続
 - <https://developer.ibm.com/recipes/tutorials/run-an-esp8266arduino-as-a-iot-foundation-managed-device/>
- 文献5: Node-REDを使ったGoogleMapへの位置表示
 - <https://flows.nodered.org/flow/1aab1d44e387da96b3fe>

• 参考資料

http://ma2.la.coocan.jp/files/toppers_ac2017/

- ソースファイル (Atollic TrueSTUDIO のプロジェクト)
- 回路図
- 動作デモ動画
- 録画AVIファイルサンプル
- 衛星測位nmeaファイルサンプル
- グーグルアースnmea表示サンプル
- IoT機能による車両位置表示サンプル