

2016 年度（平成 28 年度） 卒業プロジェクト

「自家用車におけるインターネットに接続された
車載パーソナルアシスタント HMI の研究」

“Internet-connected In-Vehicle Personal Assistant HMI (Human
Machine Interface) for Private Vehicle”

増井俊之研究会

学籍番号： 71346711 環境情報学部 4 年 名児耶均基

論文要旨

近年、希少価値の高まっている旧式のスポーツカー等を狙った盗難事件が多発しており、駐車中の車両盗難やいたずらが不安材料であるが、車と常に繋がっているような安心感があればそのような不安は払拭されと考えられる。

本研究では、自家用車におけるインターネットに接続された車載パーソナルアシスタント HMI を設計・実装した。これにより、安価でかつ手軽な手法で駐車中の不安を払拭し、擬人化エージェントとの対話を通して心理的安心感を生み出すことに成功した。

キーワード： 自動車、セキュリティ、HMI、Twitter、擬人化エージェント

慶應義塾大学環境情報学部環境情報学科 名見耶均基

目次

第一章 研究背景と目的	5
1.1. 研究背景	5
1.2. 研究目的	5
1.3. 本文書の構成	5
第二章 先行事例および関連研究	7
2.1. 先行事例・関連研究	7
2.1.1 カーセキュリティ分野	7
2.1.2 インフォテイメント分野	7
2.2. 本研究の新規性と予想される効果について	8
第三章 システムの提案と実装	9
3.1. システムの概要	9
3.2. システムの構成	10
3.3. システムの実装	12
3.3.1 <i>Twitter</i> のタイムライン監視	13
3.3.2 カーセキュリティ機能	13
3.3.3 位置情報の参照機能	14
第四章 評価	16
4.1. 評価	16
4.2. 考察	16
第五章 結論	17
5.1. 本研究によって得られた成果	17
5.2. 今後の課題と展望	17
謝辞	18
付録 A (ソースコード)	19
A.1. twistream.py	19
A.2. acceralation.py	20
A.3. location.py	21

参考文献	22
------------	----

第一章 研究背景と目的

1.1. 研究背景

近年、希少価値の高まっている旧式のスポーツカー等を狙った盗難事件が多発しており、イモビライザー¹等の盗難防止装置の付いていない車両は、特に駐車中の車両盗難やいたずらが不安材料である。

旧式のスポーツカー等を所有するような車が好きな人間にとって、マイカーは言わば家族や恋人のような存在であり、車と常に繋がっているような安心感があればそのような不安は払拭されることが考えられる。

しかしながら、市場に出回っているカーセキュリティ装置は高価であり、更に取り付けにも専門知識を要する。いたずらや窃盗行為の振動や犯人の接近を検知して警報音を出すような物が多く、万が一車両を盗難されてしまった場合は為す術がない。また、GPS を用いて愛車の位置情報を参照したり、PHS/3G 等のデータ通信回線を用いて異常を通知したりするものは一部のハイエンドモデルに留まる上、月額料金や位置情報の参照の度に料金を請求されるというのが現状だ。

小型で安価なコンピューターやセンサー類が手に入るようになっており、それらを組み合わせて応用すれば、既存の物と比較して更にスマートでインタラクティブなカーセキュリティ装置を実現できるのではないかと考えた。

1.2. 研究目的

本研究では、自家用車におけるインターネットに接続された車載パーソナルアシスタントヒューマンマシンインタフェース（HMI）を設計・実装することにより、主にいたずらや窃盗行為に対する不安を低減し、車と繋がっているような安心感を生み出すことで、より快適なカーライフを送ることが出来るようにすることが目的である。具体的には、自家用車に、センサー等を接続した小型のコンピューターを設置し、いたずらや窃盗行為を所有者に通知する。また、所有者が HMI の擬人化エージェントと対話することで、HMI の盗難通知（セキュリティ機能）をオン・オフしたり、HMI が車両の位置情報を返したりする。

1.3. 本文書の構成

本稿では、第一章にて本研究の背景と目的について述べた。第二章では、先行事例

¹ 盗難を防止するための電子的なキー照合システム

および関連研究、この研究の新規性と予想される効果について論ずる。第三章は、具体的なシステムの提案と実装方法を論述する。第四章では、評価および考察を行い、第五章では本研究の総括をする。

第二章 先行事例および関連研究

2.1. 先行事例・関連研究

本研究と関連する自動車におけるセキュリティの分野や、インフォテイメントシステムにおける擬人化エージェントの先行事例や研究を紹介する。

2.1.1 カーセキュリティ分野

市販のカーセキュリティ装置の多くは、前述のように、いたずらや窃盗行為の振動や犯人の接近を検知して警報音を出すような物に留まる。加藤電機「HORNET 370V」「NEW iVIPER」等、一部メーカーの高価なモデルには、GPS と PHS/3G 等のデータ通信回線を用いた、異常の通知機能・位置情報の参照機能を搭載したものがある。

また、警備会社の提供する「ココセコム」と呼ばれるサービスにも、愛車に機器を取り付けて位置情報を参照できるものがある。

いずれにしても通信サービスの月額使用料や位置情報取得にも都度料金がかかる。また、機器の取り付けにも専門的な知識と複雑な配線作業が求められるため、導入が難しいのが難点である。

2.1.2 インフォテイメント分野

スマートフォンやパーソナルコンピュータの分野で Apple の「Siri」や Microsoft の「Cortana」のような擬人化された一般消費者向けボットが増えているように、自動車のインフォテイメントシステムの分野においてもパーソナルアシスタント機能が増えている。例えば、BMW のインフォテイメントシステムである「iDrive」には、「スピーチ・コントロール」機能があり、自然言語処理や AI を利用したナビゲーションの設定や音楽の選曲、タイヤの空気圧やオイルの状態などの車両コンピューターの情報を参照することも可能ⁱⁱとなっている。また、アフターマーケットの商品では、ユピテル社の GPS&レーダー探知機「Lei03」では、「霧島レイ」というキャラクターが音声案内を通して、ゲーム感覚で安全運転をサポートするⁱⁱⁱという商品も販売されている。

このような、デジタルボットを介した対話型のユーザー体験が増えているのは、デジタルボットが機械の操作を簡単にするだけでなく、まるで人物とコミュニケーションをとっているような体験を作り出すからである。愛知教育大学の齋藤ひとみが、「人物的な特徴を持ったキャラクターである擬人化エージェントを用いることで、ユーザに対してより魅力的に説明や説得ができる可能性がある」と期待されている^{iv}と述べるように、味気のない機械に「人間らしさ」を付加することで、便利さと親近感を生み出す手法と言えよう。

2.2. 本研究の新規性と予想される効果について

本研究の HMI では、センサーや通信モジュールを取り付けた小型軽量ボードコンピュータを車内に設置する方式を用いる。この方法では、常時電源を車両側ヒューズ等から供給するだけで良いため、既存のカーセキュリティと比較して導入が非常に簡単である。また、車両に依存しない外付けのシステムであるため、旧式の自動車にも導入が簡単であり、手軽にセキュリティを向上させる手法としての利用価値は非常に高いと考えられる。そして、近年増えている MVNO の格安 SIM カードや、「SORACOM Air」等の IoT デバイス向けのデータ通信 SIM を活用すれば、前述のような既存のカーセキュリティと比較して位置情報の取得にかかる通信料金を安価に実現することができるだろう。

機械的なプロダクトを擬人化したキャラクターのイメージがあると愛着が湧くというのは音声合成ソフトウェア VOCALOID「初音ミク」の成功を見ても明らかである。また、自動車の車種やパーツの擬人化は、ゲームアプリ「車なごコレクション」や、トヨタの「PRIUS! IMPOSSIBLE GIRLS プロジェクト」等のように商業的に展開されるまでになっているが、特定の人間の愛車自体を擬人化したものはイラスト投稿サイトの pixiv などに少数見られる程度であり、愛車の擬人化キャラクターがどのような心理的効果をもたらすのかも調査すべき点であろう。

Twitter 等の SNS プラットフォームを介して自分の車と対話する HMI はかつて存在せず、新規性が高い。この方式には、サーバーを自分で用意することが必要ないというメリットもある。自動車という機械を擬人化し、インターネットと接続し、いつでも車両の位置情報やいたずら・窃盗行為の兆候を知ることのできる機能を持たせることで、ユーザーの不安の低減とマイカーへの愛着心を生み出し、より快適で安心感のあるカーライフを送ることが出来るようになると本研究では期待している。

第三章 システムの提案と実装

3.1. システムの概要

このシステムでは、自家用車の車内に小型軽量ボードコンピュータを設置し、愛車に異常があれば、その旨をインターネット経由で持ち主に通知し、GPS で取得した車両の位置情報を参照できるようにする。

システム（ボット）と持ち主間の対話には **Twitter** を使用し、加速度センサーを用いた車両のいたずら・盗難の兆候を検知する機能のオン・オフ、および位置情報の参照をリプライのツイートによって実現する。

今回の **HMI** ではマイカーを擬人化し、パーソナルアシスタントとして愛車の様子を見守るという設定にした。擬人化のイメージをわかりやすくするために、オリジナルキャラクターを用意した。著者の名前である「均基（まさき）」と、愛車である「マツダ・RX-8」のエイトを振り、「真咲エイト」と命名した。以下がそのキャラクターと著者の愛車の写真である。



図 3.1: 著者の愛車をイメージしたオリジナルキャラクターである「真咲エイト」



図 3.2: 著者の愛車である「マツダ・RX-8」

3.2. システムの構成

ハードウェアのメインボードには、小型軽量ボードコンピュータの「Raspberry Pi 3 Model B」を採用した。インターネットに常時接続するために Huawei 製の USB 接続の 3G データ通信モジュール「EMOBILE GD03W」を用いた。SIM カードには株式会社インターネットイニシアティブの「IIJmio データ通信専用 SIM」を用いた。これにより、月額税抜き 900 円での運用が可能になる。

車両の位置情報を取得するための GPS モジュールには、U-blox 社の「NEO-6M」を搭載した UART 通信が可能なモジュールを使用した。また、受信感度を向上させるために GPS モジュールに汎用のアクティブアンテナを接続した。Raspberry Pi と GPS モジュールの通信には GPIO の UART のシリアル通信を用いている。

車両のいたずら・盗難の兆候を検知するための加速度センサーには、3 軸加速度センサーモジュールの「ADXL345」を使用した。Raspberry Pi と加速度センサーモジュールの通信には I²C のシリアル通信を用いている。

ソフトウェア環境には、Raspberry Pi Foundation の公式オペレーティングシステムである「Raspbian」の最新版、「Raspbian Jessie with PIXEL」を用いた。

そして、ボットプログラムの開発には Python 言語を用いている。

ハードウェアへの電源供給は、車両の車内ヒューズボックスからエーモン製「【1542】電源ソケット(ヒューズ電源タイプ)」および、汎用の USB シガーソケットチャージャーを用いて行った。

システムの構成図は以下のようにになっている。

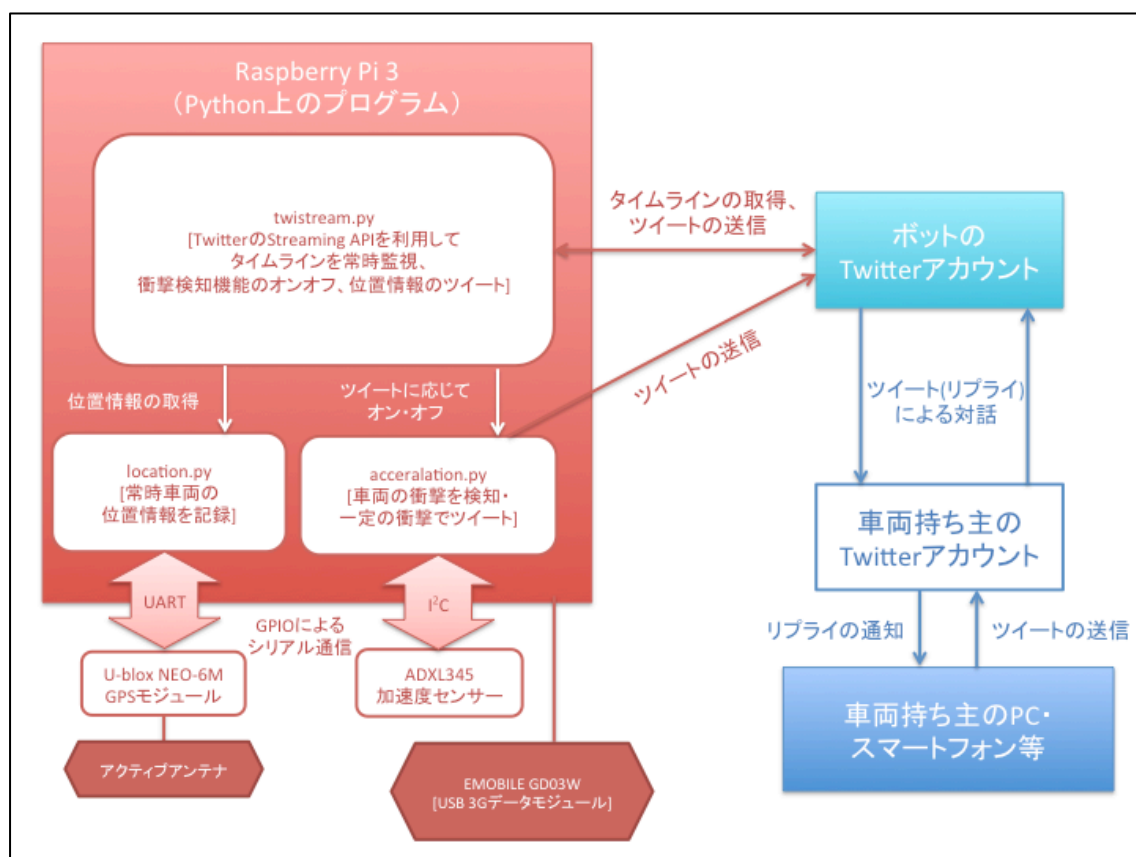


図 3.3: システム構成図

なお、実際の HMI は以下の写真のようにになっている。



図 3.4: 助手席グローブボックスに設置している HMI のハードウェア



図 3.5: ダッシュボードに設置した GPS 受信感度強化用の汎用アクティブアンテナ

3.3. システムの実装

まず、HMI ボット用の Twitter のアカウントとして@raspi_car_sec を作成した。



図 3.6: HMI の利用するアカウント「真咲エイト (@raspi_car_sec)」の Twitter アカウント

ボットの機能として 3 つの大きな機能を用意した。以下が各機能の詳細と実装方法である。

3.3.1 Twitter のタイムライン監視

今回のシステムでは、Twitter を介してユーザーと対話するため、Twitter の Streaming API を用いて常時リプライの監視を行っている。それを担うプログラムとして、「twistream.py」というメインプログラムを実行している。Python 上で Twitter を扱うライブラリとして「TwitterAPI」(<https://github.com/geduldig/TwitterAPI>)を用いた。メインプログラムを実行すると、同時に後述の位置情報を常時記録する「location.py」を別のスレッドで実行している。

「twistream.py」では、Twitter のリプライを常時 Streaming API で監視しており、特定ユーザー（車両持ち主、今回の場合は著者の大学用の TwitterID である@knps_sfc）のリプライに含まれる特定の文字列を判定して、後述の所定の動作を行うようになっている。

3.3.2 カーセキュリティ機能

愛車がいたずらや窃盗行為の脅威にさらされた時の為の通知機能である。Raspberry Pi に接続された加速度センサー「ADXL345」を用いて、一定の加速度があれば車両の持ち主にリプライを送信する。これにより、持ち主は素早い初期対応を行うことが可能になる。持ち主がボットに「オン」、「ON」、「on」というワードの入ったリプライを送信することで、「acceralation.py」のスレッドが実行され、この機能を有効化することができる。また、持ち主が「オフ」、「OFF」、「off」というワードの入ったリプライをボットに送信することでこの機能が無効化される。機能のオン・オフの際にはその旨のツイート（リプライ）が持ち主に投稿されるようになっている。Python 上で ADXL345 を扱うライブラリには「ADXLpython」(<https://github.com/pimoroni/adxl345-python>)を使用している。ADXL345 上で 0.1 秒の間に 0.05G の加速度変化があると持ち主に通知するようにした。これは、センサーの誤検知がなく、ドアを開閉する程度の揺れを取得することが出来る適切なしきい値を測定して決定したものである。



図 3.7: カーセキュリティ機能をオンにした際のリプライツイート



図 3.8: 車両に異常があった際のリプライツイート



図 3.9: カーセキュリティ機能をオフにした際のリプライツイート

3.3.3 位置情報の参照機能

「twistream.py」の実行時に自動起動される「location.py」により、Raspberry Pi に接続された GPS モジュール「NEO-6M」の位置情報を 30 秒毎にテキストファイルに書き出している。Raspberry Pi 上で GPS を扱うアプリケーションとして「GPSD」(<https://savannah.nongnu.org/projects/gpsd/>) を利用した。「twistream.py」で記録された位置情報を読み込み、持ち主から「位置」、「どこ」というワードの入ったリプライが来ると、位置情報の埋め込まれたツイートを時間と共に持ち主宛に呟く。位置情報のツイートには Twitter Rest API エンドポイントのパラメーターである、「lat」に緯度の情報を、「long」に経度の情報を、「display_coordinates」に「true」をそれぞれ付加することで正確な位置情報を投稿している。



図 3.10.1: 位置情報を参照した際のリプライツイート



図 3.10.2: 図 3.10.1 の赤枠をタップし、位置情報の詳細をツイッタークライアントで確認した様子。車両の位置がピンで示されている。

第四章 評価

4.1. 評価

本研究の HMI のシステムを著者が実際にテストした結果、いつでも自分の愛車の異常を知ることが出来ること、そして位置情報の確認が出来ることによる確かな心理的な安心感を得ることが出来た。

今回の HMI のハードウェアの制作にかかるコストは、ボードコンピューターや各種センサー、モジュール等を合わせても 15,000 円弱で入手することが出来た。(中古パーツを含む) そのため、市販のカーセキュリティシステムよりも導入にかかるコストが安いことが分かった。そして、配線作業も複雑ではないため、車両への設置も 10 分程度で終わらせることが出来た。

セキュリティの機能をオン・オフする度と、位置情報を確認する度に、オリジナルキャラクターである「真咲エイト」に和まされた。擬人化された自分の愛車だけのオリジナルキャラクターがアシスタントをしてくれるという優越感と人間らしさから由来する安心感があった。

Twitter を用いてボットと対話する手法は、ボットアカウントに特定のキーワードを含めたリプライを送るだけで良いため、専用のアプリケーション等が必要なく、著者のような Twitter のヘビーユーザーには使いやすい方法であった。また、Twitter のクライアントの通知によって車両の異常がプッシュ通知で分かることも非常に有益であった。

また、GPS の位置参照機能を用いることで、広い駐車場で愛車の場所を知ることができるのも便利であった。

4.2. 考察

本装置を用いることで、常に愛車の位置情報や異常を知ることが出来るため、万が一、愛車の盗難やいたずらをされてしまった際に素早い初期対応が出来ることは非常に魅力的かつ有用であろう。しかしながら、毎回車両を離れる際と戻ってくる際にセキュリティ機能を手動でオン・オフすることが少し面倒であったのは事実であり、改善すべき点であった。

前述のように安価で導入の簡単なカーセキュリティとしては非常に有効な手段ではあるが、HMI の存在が車外から見て分かるものではないため、いたずらや窃盗を行う者に対する抑止力にはなっていなかった。異常時のアラーム等の警報があると、被害の低減になったのではないかと感じた。

第五章 結論

5.1. 本研究によって得られた成果

安価で手軽に実装できるカーセキュリティとしての小型ボードコンピュータの車載は効果的である。

窃盗などを物理的に防ぐことは出来ないが心理的安心感の効果が一番大きかった。

Twitter を用いたシステムとの対話はシンプルで使いやすく、プッシュ通知の機能もあるため、スマートフォン等のモバイルデバイスを利用する時代の HMI との対話手段として有用であると考えられる。

単なるカーセキュリティは機械的で味気ないが、擬人化したボットとのコミュニケーションを通すことで、ユニークかつ魅力的なものとなった。愛車を擬人化した自分専用の（パーソナライズされた）キャラクターは、アバターのように特に愛着を持つことができる。自分専用の擬人化エージェントとしての車載 HMI は、安心感と優越感があることが分かった。

5.2. 今後の課題と展望

本研究の HMI では予め用意された単純なメッセージの対話しかできないため、機械学習や AI の機能を擬人化エージェントに組み込み、更にリッチで便利な対話をする事が出来れば、ユーザーはもっと楽しめるのではないかと感じた。また、車両側のボードコンピュータ等と統合し、様々な情報を参照したり、車両のあらゆる操作（例えば車外からエアコンをオンにしたり、ナビゲーションの目的地をセットする等）が出来たりするとユーザーも便利であろう。

今回の実装では、愛車にいたずらや盗難の兆候があるときのみボットから自発的にリプライを送るが、例えば、一定時間以上車に乗らないと定期的に愛車が自分にメッセージを送ること等があれば、更なる愛車との一体感が生まれ、もっと車に乗りたいという気分させるのではないかと感じた。

前述のように手動でのセキュリティ機能のオン・オフが面倒であったため、Bluetooth の Beacon を用いた近接検知での自動的なセキュリティのオン・オフが出来れば更に便利で快適だったと考えられる。

謝辞

増井俊之教授、そして増井俊之研究室の院生、学部生の皆様方には大変お世話になりました。本研究以外にもあらゆるアドバイスを賜りましたことを、心から感謝しております。

また、「真咲エイト」の擬人化に協力していただいた、春野すずらん(@latte_suzuran)氏、本当にありがとうございました。

付録 A (ソースコード)

A.1. twistream.py

```
1  # -*- coding: utf-8 -*-
2
3  from TwitterAPI import TwitterAPI
4  import os
5  import datetime
6  import time
7  import threading
8  from adxl345 import ADXL345
9  from acceralation import Accera
10 from location import GetLocation
11
12 CONSUMER_KEY =
13 CONSUMER_SECRET =
14 ACCESS_TOKEN_KEY =
15 ACCESS_TOKEN_SECRET =
16
17 api = TwitterAPI(CONSUMER_KEY,
18                  CONSUMER_SECRET,
19                  ACCESS_TOKEN_KEY,
20                  ACCESS_TOKEN_SECRET)
21
22 def Stream():
23     r = api.request('user')
24     for msg in r:
25         now = datetime.datetime.today().strftime("%Y/%m/%d %H:%M:%S")
26         ripu = msg['user']['screen_name'] if 'user' in msg else msg
27         naiyou = msg['text'] if 'text' in msg else msg
28         print(naiyou)
29         if ripu == 'knps_sfc' and u'オン' in naiyou or 'ON' in naiyou or 'on' in naiyou:
30             a = Accera()
31             t = api.request('statuses/update',{'status': '@knps_sfc セキュリティをオンにしたよ ! '+now})
32             print('セキュリティがオンになりました')
33         if ripu == 'knps_sfc' and u'オフ' in naiyou or 'OFF' in naiyou or 'off' in naiyou:
34             a.stop()
35             t = api.request('statuses/update',{'status': '@knps_sfc セキュリティをオフにしたよ ! '+now})
36             print('セキュリティがオフになりました')
37         if ripu == 'knps_sfc' and u'位置' in naiyou or u'どこ' in naiyou:
38             print(naiyou)
39             f = open('ido.txt','r')
40             g = open('keido.txt','r')
41             ido = f.read()
42             keido = g.read()
43             t = api.request('statuses/update',{'status': '@'+ripu+u' ここにいるよ ! ! '+now, 'lat':str(
44                 ido), 'long':str(keido), 'display_coordinates': 'true'})
45             print(now)
46             print(t.status_code)
47
48 def TwiStream():
49     while True:
50         try:
51             Stream()
52         except:
53             print("Unable to connect to Twitter. Wait a minute and reconnect...")
54             time.sleep(60.0)
55             continue
56
57 threading.Thread(target=GetLocation).start()
58 TwiStream()
```

A.2. acceralation.py

```
1  # -*- coding: utf-8 -*-
2  from TwitterAPI import TwitterAPI
3  from adxl345 import ADXL345
4  import time
5  import datetime
6  import threading
7
8  CONSUMER_KEY =
9  CONSUMER_SECRET =
10 ACCESS_TOKEN_KEY =
11 ACCESS_TOKEN_SECRET =
12
13 api = TwitterAPI(CONSUMER_KEY,
14                  CONSUMER_SECRET,
15                  ACCESS_TOKEN_KEY,
16                  ACCESS_TOKEN_SECRET)
17
18 class Accera():
19     def __init__(self):
20         self.stop_event = threading.Event()
21         self.thread = threading.Thread(target = self.target)
22         self.thread.start()
23
24     def target(self):
25         adxl345 = ADXL345()
26         axes = adxl345.getAxes(True) #initialize
27
28         time.sleep(5.0)
29
30         while not self.stop_event.is_set():
31             adxl345 = ADXL345()
32             axes = adxl345.getAxes(True)
33             xold = axes['x']
34             yold = axes['y']
35             zold = axes['z']
36
37             time.sleep(0.1)
38
39             axes = adxl345.getAxes(True)
40             x = axes['x']
41             y = axes['y']
42             z = axes['z']
43
44             xdif = x-xold
45             if xdif < 0:
46                 xdif = -xdif
47
48             ydif = y-yold
49             if ydif < 0:
50                 ydif = -ydif
51
52             zdif = z-zold
53             if zdif < 0:
54                 zdif = -zdif
55
56             now = datetime.datetime.today().strftime("[%Y/%m/%d %H:%M:%S]")
57
58             if xdif > 0.1 or ydif > 0.1 or zdif > 0.1:
59                 print "Detected Acceralation!"
60                 t = api.request('statuses/update',{'status': '@knps_sfc 車両の異常を検知したよ!(>w<) '+now})
61                 time.sleep(60.0)
62
63             print('実行中')
64
65     def stop(self):
66         self.stop_event.set()
67         self.thread.join()
```

A.3. location.py

```
1  #! /usr/bin/python
2  # -*- coding: utf-8 -*-
3  import os
4  from gps import *
5  from time import *
6  import time
7  import ConfigParser
8
9  def Location():
10     session = gps()
11     session.stream(WATCH_ENABLE)
12
13     keido=""
14     ido=""
15     for report in session:
16         if report['class'] == 'TPV':
17             keido = report['lon']
18             ido = report['lat']
19
20             f = open('keido.txt','w')
21             f.write(str(keido))
22             f.close
23
24             g = open('ido.txt','w')
25             g.write(str(ido))
26             g.close
27
28     print(ido, keido)
29     time.sleep(30.0)
30
31 def GetLocation():
32     while True:
33         try:
34             Location()
35             print('実行中')
36         except:
37             print('Waiting 60 sec, then reconnect GPS...')
38             time.sleep(60.0)
39             continue
```

参考文献

齋藤ひとみ. (2015). 擬人化エージェントによるオーバーハードコミュニケーション : 被説得エージェントの反応の違いについての検討. 著: 愛知教育大学, 愛知教育大学研究報告. 教育科学編 (64 輯, ページ: 141-145).

i 50 億ドル市場の人工知能、Amazon Alexa、Microsoft Cortana、Apple Siri が

ii ここまで進歩した！BMW7 シリーズの音声認識がすごい理由 | エコカー大戦争！ | ダイヤモンド・オンライン [<http://diamond.jp/articles/-/87944>]

iii Lei03 - GPS&レーダー探知機「霧島レイ」モデル
[<http://lei-kirishima.jp/sp/lei03/>]

iv 上記文献