

A. Internet of Things

1. 目的

コンピュータ・スマートフォンといった情報・通信機器だけでなく、日常生活で用いるさまざまな「物体(モノ)」に通信機能を持たせることで、インターネットに接続したり相互に通信したりするという **Internet of Things (IoT・モノのインターネット)** という考えかたが広まってきている。その実現方法の1つとして、マイクロコントローラ(マイコン)と呼ばれる小さなコンピュータをモノにとりつけ、センサによりそのモノに関する情報をマイコンに取り込み処理する方法が挙げられる。

本演習では、マイコンを用いた **Internet of Things** の実現方法について学習し、班ごとに **Internet of Things** の考えかたを用いてなんらかの課題を解決するシステムやサービスを考案し、構想→設計→実装までをひととおり実施する、プロジェクトベースの演習を行う。演習では、システムの実装に加えて、どのような問題を解決するのかの構想発表と最終的に完成したシステムがどのように役に立つのかのプレゼンテーションを行う成果発表も含み、技術の習得だけではなく、企画構想力・計画実行力の養成と分担・協力の実践を目指す。

※アイデアの考案も含めると、班に貢献する方法はたくさんあるので、例えば、プログラミングが苦手だという人は、アイデアをたくさん出すとか、はんだ付けをがんばるとか、プレゼンテーションを準備するとか、なんらかの形で班に貢献することを考えること。

2. センサとマイクロコントローラ

センサとは、なんらかの物理現象を電気信号に変換する素子のことである。加速度・距離・圧力などさまざまなものがあるが、計測した結果は電圧や抵抗や電流の変化として出力される。抵抗・電流の場合は、後述する方法により電圧に変換され、この電圧の値をマイクロコントローラ(マイコン)で取得することによって、システムは環境情報を取得する。マイコンは、通常のコンピュータが様々な用途で用いられ、汎用コンピュータと呼ばれているのは異なり、専用の処理を行う小さなコンピュータである。

センサが物理現象を計測する原理はセンサによりさまざまであるが、多くのセンサは、3

つ足か2つ足である。3つ足のセンサは、通常1つが電源電圧・1つがGND・1つが信号出力であり、物理量の変化を直接電圧の変化として読みとることができる。具体的には、電源とGNDを適切につなぎ、信号出力をマイコンの入力ポートにつなげば、値を読むことができる。2つ足のセンサは、物理量が変化することによって、内部抵抗が変化したり電流を発生させたりする。この場合は、以下に述べるプルアップ・プルダウンという方法で、抵抗・電流の変化を電圧値として読むのが一般的である。

プルアップ・プルダウンとは、図1に示すとおり、電源電圧やGNDからマイコンの入力端子に抵抗をはさみこむことにより、内部抵抗の変化を分圧の法則により、あるいは、発生電流を電圧降下の原理により、電圧の変化に変換する回路である。また、スイッチやセンサが接続されていない場合や稼働していない場合でも、マイコンに適切な電圧が入力されるようにする働きもある。はさみこむ抵抗のことをプルアップ抵抗・プルダウン抵抗と呼ぶ。図1において、スイッチが切れている場合やセンサの内部抵抗が十分に大きい場合は、5Vと0Vの間に電流は流れないため、プルアップ抵抗・プルダウン抵抗における電圧降下は発生せず、出力線にはプルアップの場合5V・プルダウンの場合0Vが出力される。一方、スイッチを入れるかセンサの内部抵抗が小さくなると、プルアップ抵抗・プルダウン抵抗に電流が流れ、その電圧降下により出力線の電圧が変化する。センサが電流を発生させるタイプの場合は、その電流がプルアップ抵抗・プルダウン抵抗に流れることによる電圧降下で、出力線の電圧が変化する。周辺の回路により、プルアップがよい場合とプルダウンがよい場合があるが、本演習の範囲ではどちらでも大きな差はない。プルダウンのほうがセンサ入力値の上昇として読みとることとなり、直感的である。

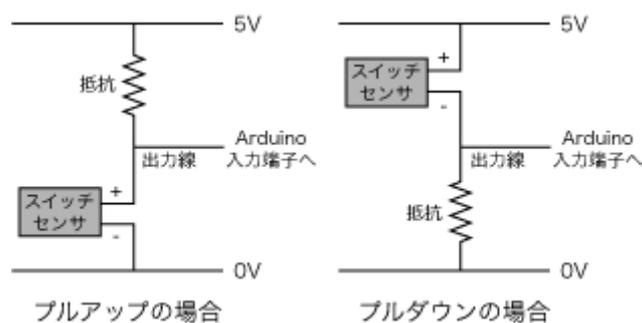


図1 プルアップ回路・プルダウン回路

本演習では、以下のセンサが使用可能である。付属するデータシートを読むと、どのような回路により物理量の変化をとらえることができるか書いてあるので、データシートと上記の説明をよく読み、どのような回路が必要になるのかを考えること。これらのセンサを用いてどのようなモノの状態を計測できるかを考え、実装するシステムを考案すること。また、これ以外のセンサ・部品でも担当教員が所有している場合があるため、システム実装のため必要なものがある場合には担当教員に相談すること。

- ・ 加速度センサ

モノの動きを感知できるとともに、重力方向もわかるため姿勢を認識することができる。

<https://www.switch-science.com/catalog/1970/>



- ・ 距離センサ

モノまでの距離を測ることができる。設置のしかた次第で、位置関係を知ることができる。

<http://akizukidenshi.com/catalog/g/gI-07546/>



- ・ 圧力センサ

人間がふれていることが検知できる。例えば椅子にとりつければ、座っているかどうか分かる。また、ユーザの入力に使うことができる。

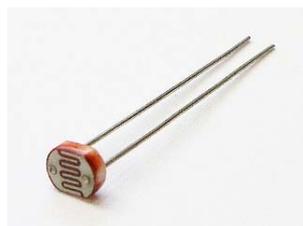
<http://akizukidenshi.com/catalog/g/gP-04003/>



- ・ 照度センサ

電気がついた・日が出ているなど環境の情報を取得できる。

<http://akizukidenshi.com/catalog/g/gI-00110/>



マイコンを動作させるには、マイコン上で動作するプログラムを開発しなければならないが、汎用コンピュータとは異なり、マイコン自身に開発環境は備わっていないため、汎用コンピュータでマイコンのプログラムを作成・コンパイルし、プログラムをアップロードする手順が必要となる。そのプログラムには、センサから出力された電圧値を読み、その値を外部に通信する処理が最低限必要である(図 2)。本演習では、**Internet of Things** の第 1 歩として、マイコンと汎用コンピュータを USB による有線をつなぎ、汎用コンピュータでマイコンからの値を受けとって動作するシステムを設計・実装することを基本課題とする(図 2)。(早く進んだ班は、通信の無線化やスマートフォンへの出力にも取りくむことを期待している。)

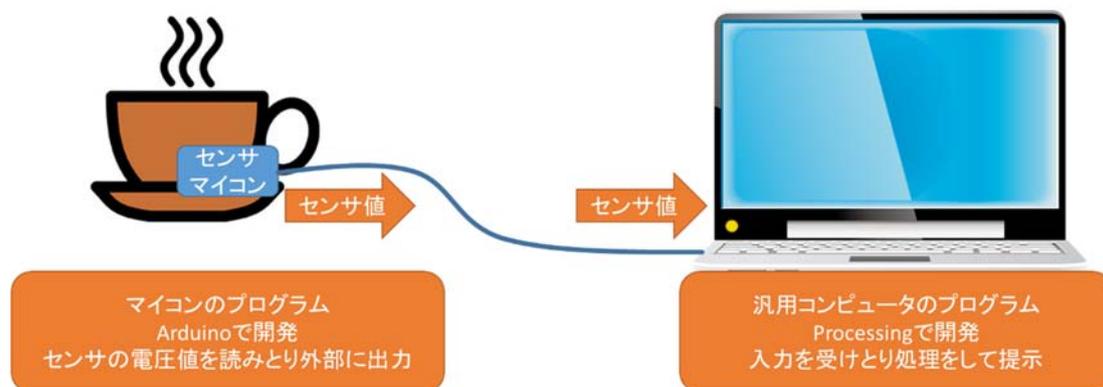


図 2 マイコンと汎用コンピュータの通信

本演習は、マイコン開発環境として **Arduino** を用意しており、汎用コンピュータ上で動作するアプリケーションの開発環境として **Processing** を用意している。どちらも未経験者でも容易に開発を行うことができるよう十分考慮されており、3 週しかない本演習での使用にも向いている。また、**Processing** には **Android** モードがあり、**Android** アプリに容易に移植可能である。

3. Arduino

Arduino は、**Atmel** 社の **AVR** マイコンを使用したマイコンボードである(図 3)。また、マイコンのプログラムを作成するための **Arduino** 言語とその開発環境である **Arduino IDE** も用意されており、**Arduino IDE** でプログラムの作成・コンパイル・アップロードまでの一連の作業を行うことができる。**Arduino IDE** の外観は、プログラムを編集するためのエディタ部分と上部にボタンが並ぶ構成となっている(図 4)。エディタ部分でプログラムを編集し、コンパイル・アップロードボタンを押すことによって、**USB** でつながったマイコン

ボードにプログラムが書きこまれる。また、シリアルモニタボタンを押すことで、マイコンからの出力を確認することができる。



図3 Arduino マイコンボード



図4 Arduino IDE の外観

Arduino 言語は、C++言語をベースにマイコン用の独自の関数群を備えた言語である。しかし、後述する Processing 言語の影響を強く受けており、Processing 言語からの機能の採用も見られる。特に、`setup()` と `loop()` の2つの関数は独特であり、C言語のような `main()` 関数からプログラムが開始するのではなく、`setup()` 関数がプログラムの開始時に1度だけ呼び出され、その後 `loop()` 関数が繰り返し呼び出され続けるというモデルになっている。

演習時に電圧を読みとりデータを出力するサンプルプログラムを配付するので、まず 1 つのセンサによる値の変化を確認してみるとよい。

4. Processing

Processing 言語は、Java 言語をベースにした画面描画処理を容易に実現することを目指して設計されたプログラミング言語であり、通常その開発環境である Processing Development Environment (PDE) でプログラムの作成・コンパイル・実行までの一連の作業を行うことができる。PDE の外観は、プログラムを編集するためのエディタ部分と上部にボタンが並ぶ構成となっている(図 5)。エディタ部分でプログラムを編集し、実行ボタンを押すことによって、コンパイル・実行がひとつづきに行われる。Processing 言語は、描画のための組み込み関数を多く備えており、また、アニメーションを実現するために、設定した時間間隔で繰り返し呼び出される `draw()` 関数を中心にソフトウェアを設計するビジュアルプログラミングを容易にするモデルを採用している。`setup()` 関数がプログラムの開始時に 1 度だけ呼び出され、その後 `draw()` 関数が設定した頻度(デフォルトは 60 回/秒)で繰り返し呼び出され続ける。演習時に入力されたデータをグラフ化するサンプルプログラムを配付するので、まずマイコンからのデータをグラフにより可視化してみるとよい。

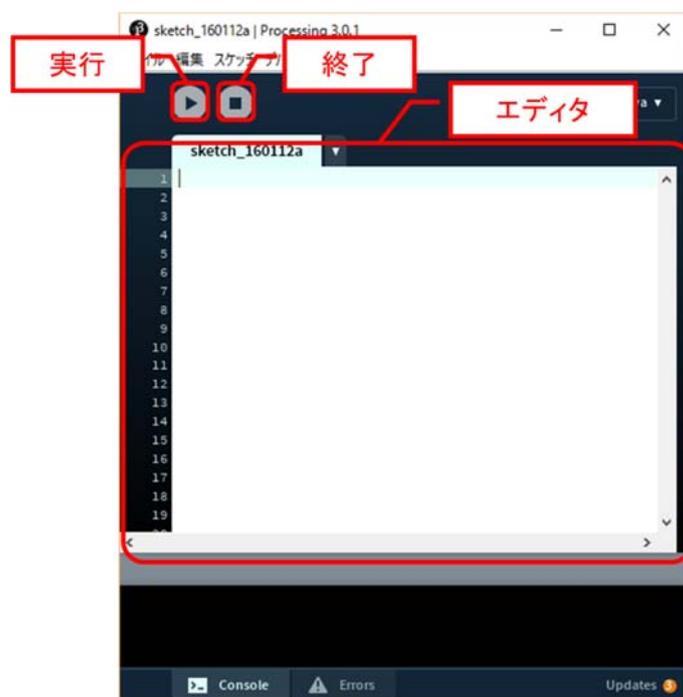


図 5 Processing Development Environment (PDE) の外観

5. 考察

(a)演習で作成したシステムに対するさらなる改善には、どのようなことが考えられるか？
その方策を可能な限りの具体的に述べよ。

(b)コンピュータの内部で動作するプログラムとは異なり、センサを扱うシステムには機械的・電氣的ノイズによる不安定さが伴う。センサの不安定さを取り除く、ハードウェア的手法かソフトウェア的手法のいずれかを考えよ。ハードウェア的手法の場合は、電気回路を示して説明すること。ソフトウェア的手法の場合は、プログラムコードを示して説明すること。

(c)Bluetooth を用いて、マイコンとスマートフォンを直接つなぐには、シリアルポートのオープンではなく、Bluetooth の直接的な操作が必要である。具体的には、アダプタの取得・サービスプロファイルの指定・ソケットの取得・ソケットのオープンという手順を踏む。その方法をプログラムコードで示せ。

6. レポート作成について

レポートは指導書冒頭の注意事項にしたがって作成すること。ただし、方法の章に「班員の役割分担」、結果の章に「最終成果物」を明記すること。システム構成を記述する際に、構想したが未完成であった部分を含んでかまわない(ただし、未完成部分は明記すること。)ので、わかりやすい記述を心がけること。考察の場合は、「班員の役割分担」・「最終成果物」・「考察」を書く。

「班員の役割分担」について

まず班員全員の役割分担を記し、次に自分が担当した項目を詳細に述べる。

「最終成果物」について

実装した成果物の写真を載せ、動作説明を行う。例えば画面遷移など、説明に必要であると思われる場合は、写真や図を効果的に用いること。