

IoT センサノード向けマイコン学習教材の開発

Development of Microcomputer Learning Materials for IoT Sensor Nodes

仲谷 茂樹 *1

NAKAYA Shigeki

要約 IoT システムを構成する要素にセンサノードと呼ばれる端末装置がある。このセンサノードは一般的にバッテリー駆動であり使用期間も比較的長期にわたるケースが多いため、端末の低消費電力化が必要となっている。これを実現するためにコントローラとして低消費電力対応のマイコンが求められており、各社より提供されているが、これらのマイコンを活用するには低電力化技術の理解が必要である。本稿は具体的な低消費電力マイコンを例とし簡易的な IoT システムの製作を通して低消費電力マイコンの活用技術を習得する教材の報告である。

1 はじめに

最近、製造業や農業、自然環境モニターなどに IoT システムを導入する事例が多くみられるようになった。当機構が実施する在職者向け訓練においても新規コースとして IoT 関連のカリキュラムコースが整備されている。一方、実際の運用において IoT におけるセンシングをおこなう端末装置（以下センサノードという）ではバッテリー駆動が一般的でありメンテナンスコストを考慮すれば、低消費電力化が必須¹⁾ となっている。

そこで今回は IoT 向けの低消費電力マイコンを使用し、IoT センサノード向けの低電力化に対応したマイコン活用技術を習得する学習教材の作成をおこなったので報告する。

2 学習教材の概要

学習教材の内容を検討するにあたり、在職者向け訓練コースの 12 時間もしくは 18 時間で使用することを想定する。なお、当機構で実施される訓練では実習の割合が 6 割以上と設定されていることから、実習ではマイコンボードとブレッドボード等を使用し、自ら簡易的な IoT システムを構築しながら低消費電力化マイコンの活用技術を習得できるようにする。

実際の IoT システムではセンサノードは複数個使用するケースが多いが本教材ではセンサノードは 1 個とし、センサ情報の受信は PC 上にておこなう。

センサとしては変化量が分かりやすく、取り扱いが容易であることを考慮し温湿度センサを使用し、一定時間間隔(60 秒)にてデータを送信する。なお、送信時にはセンサ情報取得日時を付加して送信する。

また、低消費電力化の効果を確認するためにマイコンの動作電流をモニターする必要がある。そのため容易に動作電流が測定できるマイコンボードを選定し、携帯型マルチメータ等にて測定可能な構成とした。

3 使用機器について

3-1 マイコンボード

低消費電力対応マイコンは各社より提供されているが、今回の教材では入手性およびプログラム開発環境、動作電流測定の容易さ、導入コストを考慮し ST マイクロエレクトロニクス社製の超低消費電力対応マイコン²⁾ STM32L0 シリーズが実装された評価ボードである NUCLEO-L053R8 を使用することとした。本ボードの外観を図 1 に示す。

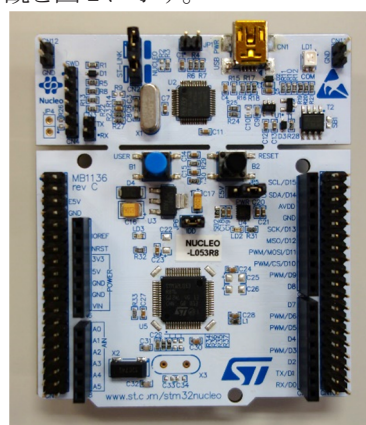


図 1 使用するマイコンボード

CPU コアは 32bit の RISC コアである ARM CortexM0+を採用し、最大動作周波数は 32MHz であ

*1 電子情報技術科

Department of Electronics and Information Engineering

る。動作電圧範囲は 1.8~3.6V となっているが実際にはチップ内部にメイン用電源レギュレータ回路と低消費電力用のレギュレータ回路の 2 系統内蔵しており、最適化した電圧にて CPU や周辺回路を動作させることが可能である。

マイコンの周辺回路としては GPIO,各種タイマー, AD コンバータ,I2C,SPI,UART, USB, DMA 等を実装している。なお、これら周辺回路へ供給するクロックも低消費電力対応のため、最適な周波数の設定が可能となっている。

さらに低消費電力動作を実現するために、以下の 5 つの動作モードをサポートする。

- ① RUN モード (CPU 動作)
- ② LowPowerRUN モード (CPU 動作・低速)
- ③ Sleep モード (CPU 停止、周辺回路動作)
- ④ Stop モード (CPU 停止、周辺回路一部停止)
- ⑤ StandBy モード (すべて停止)

①RUN モードは CPU および周辺回路が最大動作周波数まで動作するモードである。一方、②の LowPowerRUN モードでは CPU の動作周波数を抑えたモードであり、ある程度の処理時間を要しても電力を抑えたいときに使用する。③Sleep モードでは CPU を停止させ、周辺回路のみを動作させる。④Stop モードでは周辺機能回路の一部のみ動作させ、それ以外の周辺回路も停止とする。⑤StandBy モードではほぼすべての周辺機能およびメモリも停止とするものである。実習ではこれらの動作モードを設定し低消費電力の効果を確認する。

なお、マイコンボードの上部には ST-LINK/V2-1 と呼ばれるデバッグ用ハードウェアが実装されており ST マイクロエレクトロニクス社から提供されるフリーの統合開発環境 STM32CubeIDE と連携させることで C 言語でのプログラム開発を効率良くおこなうことができる。電源は USB より給電されておりボード上にあるレギュレータ回路により 3.3V が供給される。また、ボードの左右には Arduino 互換の外部拡張コネクタが実装されており GPIO をはじめマイコンの各種機能端子がジャンパー線等を使用して接続可能である。

3-2 センサ

温湿度センサとしては SENSIRION 社の高精度温湿度センサである SHT31-DIS を搭載した秋月電子通商社製センサモジュールを使用する。温度測定レンジは -40℃~+125℃(±0.2℃)、相対湿度測定 0%~100%(±2%)である。マイコンとの通信は I2C インタ

フェースであり、マスターであるマイコンからの測定開始コマンドによりモジュール内の MEMS センサにて温湿度を測定する。その後、マスターからのデータ受信コマンドによりモジュール内部で AD 変換したデジタルデータがマイコンに送信される。なお、同じく I2C インタフェースをもつ 8 文字 2 行表示可能な LCD モジュールをマイコンに接続し、センサモジュールの動作チェックやデバッグ用として使用する。

3-3 無線モジュール

IoT 向けの無線モジュールとして 920MHz 帯域の LoRaWAN や 2.4GHz 帯の ZigBee や Bluetooth 等があるが、実習時の扱いやすさを考慮して 802.15.4 準拠モジュールである XBEE (Digi International 社) を使用する。このモジュールにはワイヤアンテナが実装されており、数十メートルの距離であれば容易に無線通信が可能である。なおマイコンとのインタフェースは UART 通信を使用する。

本教材では XBEE モジュールを 2 個使用しセンサデータの送信をおこなうため、XBEE 専用ソフトを使用して使用チャンネル、PAN-ID、自分アドレス、相手先アドレス、UART の送信速度の設定をおこなう。またシステム全体での低消費電力化を行うために Sleep 機能を有効にする。これにより Sleep 端子のデジタル制御にて未送信時の待機電力を低減することができる。

PC 側に接続する受信モジュールとブレッドボード上にセンサ、デバッグ用 LCD および送信用無線モジュールを組み込んだ様子を図 2 に示す。

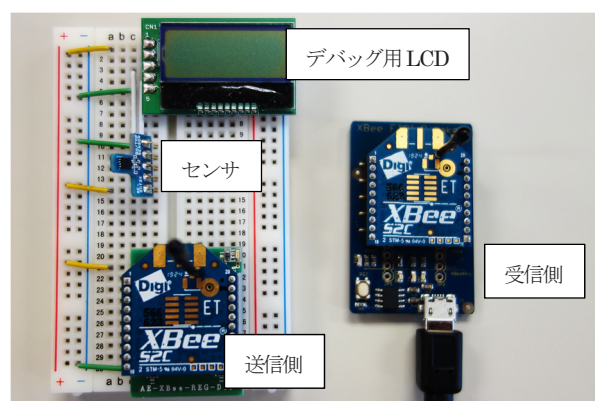


図 2 無線モジュール (送受信部)

4 実習内容

低電力化対応マイコンの制御技術を効率良く習得するために以下に示す手順で簡易的な IoT システムを構築しながら低電力化の評価が行えるようにした。

4-1 マイコン開発環境導入と GPIO 制御

実習で使用するマイコンの概要および開発環境について説明し、実際に手を動かしながら使用方法を習得する。STM32CubeIDE を使用しマイコンボードに合わせた新規プロジェクトの作成方法、マイコンシステムの各種初期設定等をおこなう。なお、動作クロック周波数の設定は STM32CubeIDE 内の Clock Configuration ツールにて詳細な設定が可能であり、低消費電力の際に重要となる。ツールにてクロック周波数を設定している様子を図 3 に示す。

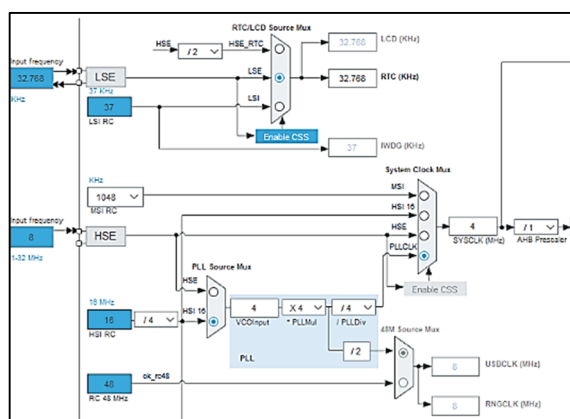


図 3 クロック周波数設定画面

最初に作成するプログラムは GPIO を使用し、マイコンボード上に実装されたプッシュスイッチと LED を点灯制御するプログラムを作成する。

またデバッグ用の ST-Link には仮想 COM ポート機能が、UART として使用することができる。マイコンのシリアル通信モジュールの USART2 を有効に設定、転送速度を 9600bps とし、固定文字列の出力プログラムを作成し動作確認をおこなう。

なお、STM32CubeIDE でのプログラム開発では HAL ライブラリの使用が前提となっている。HAL とは Hardware Abstraction Layer の略で、ハードウェアを抽象化することで様々なハードウェアからアクセスを可能とする API のことである。今回のプログラムでは HAL でのコーディングが必須のため HAL ライブラリの参照および記述方法についても適宜説明する必要がある。

4-2 I2C 通信によるセンサ制御プログラム

センサノードデバイスとして温湿度センサを I2C インタフェースにて接続する。I2C 通信の基本プロトコルおよび HAL ライブラリでの I2C の使用方法を説明

し、マスタースレーブ間のデータ送受信プログラムの作成をおこなう。なお、温湿度センサにて測定した値を同じ I2C インタフェースを持った LCD に結果を表示するプログラムを作成し正常にセンシングできているか確認することができる。

4-3 データ送信プログラム

マイコン導入部で設定した UART のプログラムを修正し温湿度データを無線モジュールである XBEE を介してホスト PC に送信するプログラムを作成する。

送信時には取得した日時情報も付加して送信する必要があるためマイコンに内蔵されているリアルタイムクロック (以下 RTC という) モジュール有効にして各種設定をおこなう。RTC 用の HAL ライブラリにより時刻設定をマニュアルにて設定したのち RTC による周期割込み (60 秒) にて取得日時と測定データを合わせて PC に送信する。PC 側ではターミナルソフトを使用して受信しデータの確認が可能である。

4-4 CPU 動作時の低消費電力化

低消費電力化の評価をおこなうためマイコンボード上の電流測定端子にマルチメータを接続する。実際に測定している様子を図 4 に示す。

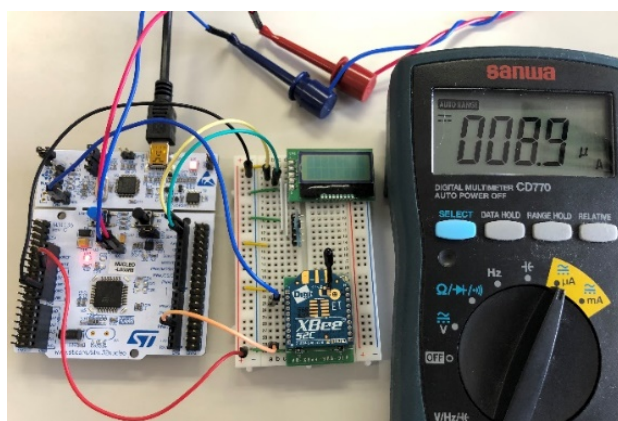


図 4 マイコンの動作電流を測定

CPU が動作した状態での低消費電力化手法として動作クロック周波数の最適化がある。統合開発環境の STM32CubeIDE で Clock Configuration ツール設定にて内蔵のクロックソースを使用し動作周波数と動作電流の関係を確認する。内蔵のクロックソースを中速内蔵クロックモード (以下 MSI モードという) で使用し 0.26MHz~4MHz まで変化させた場合の動作電流の変化の例を図 5 に示す。

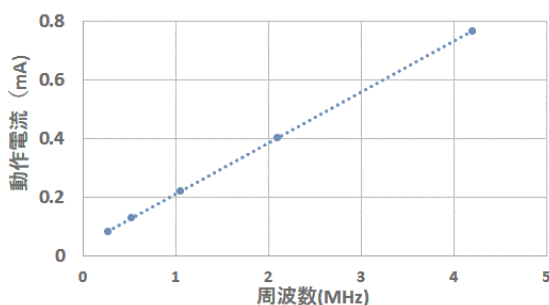


図5 周波数と動作電流の関係

グラフからわかるように周波数に正比例して動作電流が多くなるため低消費電力を実現するには最適な動作周波数を設定する必要があることがわかる。また高速処理が必要な RUN モードと低速処理のモードである LowPowerRUN をソフトウェアにて切替えることにより動作電流が低減されることを実習にて確認する。

4-5 Sleep/Stop モードによる低消費電力化

低消費電力化を実現するために間欠動作を実装する。間欠動作とは処理が必要な時だけ CPU を動作させ、それ以外は CPU を停止させることで平均の消費電流を低減する手法である。今回は時刻取得および温湿度測定、PC へのデータ送信後は CPU を完全停止とする Sleep モードと RTC 回路を除く周辺回路までを停止する Stop モードの 2 通りを実装する。なお Sleep もしくは Stop モード移行には割り込み処理が必要となり、ここでは RTC によるアラーム割り込み(60 秒)を使用する。

各モードへのエントリや割り込み処理の記述についても HAL ライブラリを利用して記述する。

CPU 動作周波数を MSI モードの 262KHz とし、間欠動作するプログラムを作成しそれぞれの待機電流を測定した結果を示す。なお、間欠動作では CPU 動作時間が 0.13 秒(動作電流は 82 μ A)、Sleep もしくは Stop による停止時間は 59.87 秒である。

- Sleep モード待機時電流 : 59 μ A
- Stop モード待機時電流 17 μ A

次に両モードにおいてチップ内部の電源レギュレータ回路を低消費電力用に切り替えることでさらに削減が可能であることを確認する。

- Sleep モード待機時電流 : 36 μ A
- Stop モード待機時電流 2.5 μ A

なお、平均消費電流は(動作時間×動作電流+停止時間×待機電流)/(動作時間+停止時間)で算出できる。

マイコンの低電力化機能をフル活用することにより平均消費電流が約 2.7 μ A となることがわかった。

なお、センサおよび送信部まで含むすべての平均消費電流値は 30 μ A 程度となり、簡易的な計算ではあるが理想のマンガン電池 (700mA/h) を直列 2 本構成で使用した場合、約 900 日以上動作が可能となる。

4-6 応用課題 (電波時計モジュールの導入)

RTC への現在時刻設定をマニュアルでなく電波時計モジュールを利用し自動にて設定する課題を応用課題とする。電波時計とは情報通信研究機構が送出している標準電波を受信することで、正確な年月日および現在時刻を取得するものである。今回はイズム社³⁾の試作標準電波受信モジュールをお借りして動作確認を行った。モジュールから出力される 1 秒周期のタイムコードをマイコンの GPIO にて取り込み、プログラムにてデコードすることで年月日時分情報が得られる。なお標準電波受信のプログラムはボリュームも多く、実動作でのデバッグに時間を要することから実習ではプログラムの一部を完成させることとする。図 6 に使用したモジュールの外観と図 7 に出力されるタイムコードの波形を示す。

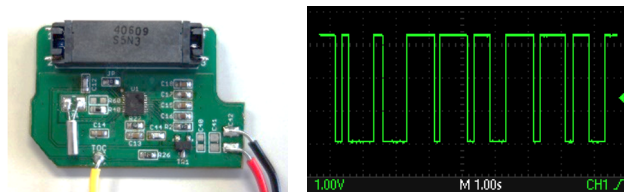


図6 電波時計モジュール 図7 タイムコード波形

5 まとめ

IoT のセンサノード向けの低電力対応マイコンの活用技術を習得するための教材を作成した。センサノードで必要となる低消費電力を実現するための制御方法を簡単な IoT システムの製作を通して習得することができる。今後は在職者訓練にて活用し、課題の見直しや改善をおこなう予定である。

6 謝辞

電波時計モジュールは宮城県の(株)イズム様に提供頂きました。ここに感謝申し上げます。

参考文献

- 1) 科学技術振興機構研究開発戦略センター, IoT 時代のセンサ融合基盤技術の構築, p24, 2020
- 2) <https://www.st.com/ja/microcontrollers-microprocessors/stm32l053r8.html>
- 3) <http://www.izm-e.com/>