

生成AIを利用したプログラミング支援システムの開発 — マイコンボードへの応用 —

天良 和男

元東京都立日比谷高等学校

元東京学芸大学

tenra.kazuo@nifty.com

川瀬 賢二

株式会社アーテック企画室

k-kawase@artec-kk.co.jp

生成AIを活用したプログラミング支援システムの多くは、PC上での入出力に留まっている。

本システムでは、OpenAI APIを利用することで、プログラミング言語に不慣れな初心者でも、アルゴリズムを記述した日本語の指示文（プロンプト）を入力するだけで、生成AIが出力したコードをPC上に留まらず、マイコンボードに直接転送・実行することができる。この機能により、ロボットや鉄道模型などの実物を制御する体験を通じて、論理的思考力を育成することが可能となる。さらに、出力調整機能により、学習者のスキルや発達段階に応じたコードやコメントの生成が可能である。不明瞭で具体性に欠けるプロンプトにはコードを生成しないよう制御できるため、安易なAI依存を防ぎ、学習者が正確かつ具体的なアルゴリズムに基づいたプロンプトを意識的に作成するよう促すことができる。なお、本システムは、マイコンボードを使わない一般的なプログラミングにも使用できる。

1. はじめに

近年、生成AI技術の発展により、プロンプトを用いたコード生成が可能となり、プログラミング教育への応用が注目されている。特に、生成AIを活用することで、プログラミング言語に不慣れな初心者でも、アルゴリズムを示す文章を入力するだけでコードを生成できるようになる。一方で、生成AIが出力したコードの実行がPC上に留まっている。また、具体性に欠けるプロンプトであっても正しいコードが生成されてしまう場合があり、それによって安易にAIへ依存し、思考力や試行錯誤する力の低下につながるおそれがある。

2. 研究目的と解決方法

生成AIが出力したコードの実行がPC上に留まっている状況を踏まえ、活用の幅を広げることを目的として、PCとマイコンボード間でコードを転送・実行できる独自のクラスを開発した。これにより、マイコンボード上で生成されたコードを実行できるようになり、ロボットや鉄道模型などの実物の動きを通して、コードと現実世界の動作との関係を直感的に理解できるようになった。

また、安易なAI依存による思考力や試行錯誤する力の低下を防ぐために、学習者のスキルや発達段階に応じたコードやコメントを生成可能とする、独自の出力調整プロンプトを導入した。これにより、不明瞭で具体性に欠けるプロンプトではコードを生成しないよう制御でき、学習者が正確かつ具体的なアルゴリズムに基づいたプロンプトの作成を意識的に行うよう促すことができた。

3. システム構成

本システムは、生成AIを活用したソフトウェア（Intelligence Software）と、マイコンボードなどのハードウェアで構成される。ユーザはテキスト入力、ファイル読み込み、音声入力、画像入力など、様々な方法でプロンプトを入力できる。PCはUSB経由でマイコンボードと接続され、生成AIが出力したMicroPythonコードを転送・実行しロボットや鉄道模型などの動作を制御できる。

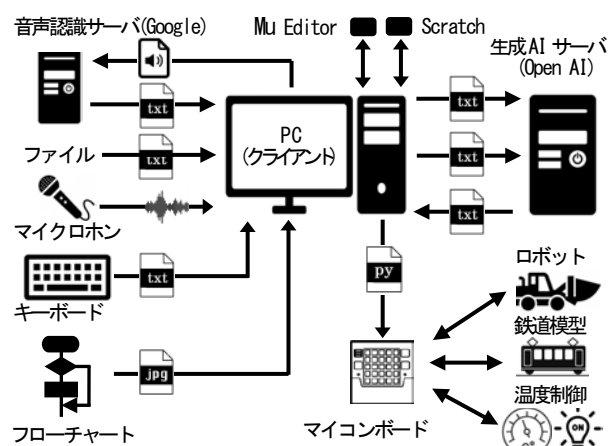


図1 システム構成図

4. プロンプト構成

本システムのプロンプトは、マイコンボードの入出力ポートやセンサ、モータ等の仕様に関する情報を設定するハードウェアプロンプトと、学習者の発達段階やスキルに応じて、生成AIが出力するコードを適切な難易度や内容に調整する出力調整プロンプト、学習者が生成AIに対する指示を設定するユーザ作成プロンプトから構成される。

5. 実験例

Intelligence Software に対応するマイコンボードは、現時点で(株)アーテックの Studuino:bit および ArtecLinks, イギリス BBC の micro:bit である。赤外線フォトリフレクタや DC モータを搭載したライントレースロボット⁽¹⁾のほか、それらにサーボモータを追加した搬送ロボットや鉄道模型などを用いて実験を行った結果、正しいコードが生成され、正常に動作することが確認された。

紙面の都合上、これらの詳細な説明は割愛するが、ここでは以下の実験例を紹介する。

図2のように、マイコンボードに拡張コネクタを接続し、そこに温度センサと緑色および赤色のLEDを取り付けた。図3の2に示すような、温度の値に応じて点灯するLEDを切り替える動作をするユーザ作成プロンプトを入力すると、同図3のコードが生成され、正しく動作することを確認した。なお、同図1に示すハードウェアプロンプトはチェックボックスを選択するだけで自動入力される。また、出力調整プロンプトはラジオボタンから選択するが、紙面の都合上説明は割愛する。

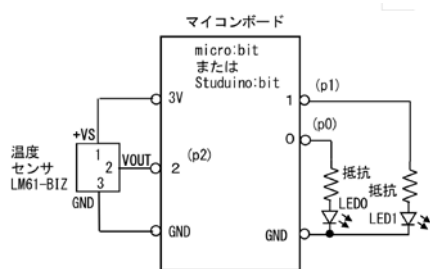


図2 温度センサ・LED回路

6. 出力調整プロンプトの違いによる出力変化

出力調整プロンプト（TYPE1～TYPE4）と、抽象度の異なるユーザ作成プロンプト（USER1～USER4）を組み合わせた実験を行い、生成AIのコード出力の変化を検証した。

USERの番号が大きくなるほど、プロンプトに含まれる情報が具体的になり、部品名や接続情報、制御メソッドの記述までが含まれる。

一方、TYPEの番号が大きくなるほど出力条件が厳格になり、部品や配線、制御方法に関する明確な情報がなければコードは生成されない。

実験の結果、生成AIに正しいコードを出力させるには、目的や使用する部品、接続方法、制御手順などを具体的かつ明確にプロンプトに記述することが重要であることが分かった。曖昧な指示では誤ったコードや実行不可能な処理が生成される可能性があり、特に出力条件が厳しい設定ではコードが生成されないこともある。そのため、学習者は自らのアルゴリズムや意図を正確にプロンプトとして記述し、表現する力を養う必要がある。

■ 1. ハードウェアプロンプト

- Pythonのバージョンは3.4です。
- 待ち時間はtime.sleep(秒単位)を使用してください。
- "from sbblocks import *"で全ての機能を制御できます。

(1) LED Display のハードウェア仕様プロンプト

- "LEDディスプレイ"は、displayオブジェクトで制御でき、次のメソッドを持ちます。
- scroll(string, delay=150, *, color): string(英数字)を水平方向にスクロールさせます。delay 引数でスクロールさせる速さ(ms)を指定できます。
- (2) Terminal (拡張端子) のハードウェア仕様プロンプト
- "端子"は、p0～p16, p19, p20 オブジェクトで制御でき、全ての端子で使用できるメソッドは次の通りです。
- write_digital(value): value 引数が1の場合はHighに、0の場合はLowにデジタル信号を設定します。
- read_analog(mv=False): 端子の電圧を読み取り、mv 引数がFalseの場合、0 (0V) から 4096 (3.3V) までの間の整数値を返します。mv=Trueの場合、mVで返します。

■ 2. ユーザ作成プロンプト

[部品の説明]

- p0に緑色のLEDが接続されている。
- p1に赤色のLEDが接続されている。
- p2に温度センサが接続されている。

[動きの説明]

温度 = 330 * 温度センサの値 / 1023 - 60
 温度が30℃以上ならば赤色のLEDが0.5秒間隔で5回点滅して終了し、そうでなければ緑色のLEDが1秒間隔で3回点滅して終了してください。温度は表示してください。

■ 3. 生成されたコード

```
<code>
import time
from sbblocks import *
sensor_value = p2.read_analog() #温度センサの値
temperature = 330 * sensor_value / 1023 - 60 #温度計算

#温度表示
display.scroll("Temp: {:.1f}C".format(temperature),
delay=150)

if temperature >= 30: #温度が30℃以上か
    for i in range(5): #赤色のLEDを5回点滅
        p1.write_digital(1) # 赤色 LED ON
        time.sleep(0.5)
        p1.write_digital(0) # 赤色 LED OFF
        time.sleep(0.5)
else:
    for i in range(3): #緑色のLEDを3回点滅
        p0.write_digital(1) # 緑色 LED ON
        time.sleep(1)
        p0.write_digital(0) # 緑色 LED OFF
        time.sleep(1)
</code>
```

図3 プロンプトと生成された回答コード

7. おわりに

本システムには、フローチャート画像の入力やPythonをScratchに変換する機能、マイコンボードを使わない一般的なプログラミングにも使用できる機能も実装しているが、紙面の都合上割愛する。今後は、多様な環境への適用や、プロンプト機能の強化など、改良を進めていく予定である。

参考文献

- (1)天良和男, 川瀬賢二: 生成AIを利用したプログラミング的思考を育成するためのマイコンボード対応プログラミング支援システムの開発, 日本情報科教育学会第24回研究会報告書, pp.13-18(2025).