

プログラミングの多様性を引き出すワンボードマイコンの活用

中尾尊洋

鳥取大学附属中学校 技術・家庭科（技術分野）

E-mail: nakaot@fuzoku.tottori-u.ac.jp

Takahiro NAKAO (Tottori University Junior High School) : Utilization of microcomputers enabling various styles of programming.

要旨 — 従来のプログラミング学習では、授業者がある程度完成されたプログラムを提示する。それを写し取らせることで、学習者に命令の組み合わせや処理について理解させる。しかし、このような学習方法は、完成されたプログラムが提示されるために、プログラミングの多様性を理解する際の障害要因となる。そこで、自ら命令を組み合わせるプログラミングできるように支援をする学習方法、及び、それを可能にする教材開発を行った。この教材を用いた授業実践において、学習者のプログラミングに多様性が見られたので、その内容について報告する。

キーワード — プログラミング教材、プログラミングの多様性、主体的学習、試行錯誤

Abstract — Teachers have presented a single program which is already completed to a certain extent in the conventional classes of the programming learning. By copying the program, students have been supposed to understand the combination and processing of the instructions. However, such learning style is disincentive for the understanding of the diversity of programming, because a completed program is presented as if it was the single answer to the problem to be solved. To avoid this problem, we have developed a learning method that supports programming by combining instructions themselves, and teaching method that enables it. Programs made by students during the lesson using this teaching method were diverse as reported in this article.

Key words — programing teaching material, variety of programming, independent learning, trial and error

1. はじめに

2018年度から実施される新しい学習指導要領に向けた議論（文部科学省、2016）の中で、「何を知っているか」ではなく、「何ができるようになるか」を重視するという視点が共有された。情報教育に関しては、「世の中の様々な事象を情報とその結びつきとして捉えて把握し、情報及び情報技術を適切かつ効果的に活用して、問題を発見・解決したり自分の考えを形成したりしていくために必要な資質・能力」を情報活用能力と捉え、その育成が求められている。プログラミングに関しては、「プログラミング的思考」を育むプログラミング教育について述べられている。

中学校におけるプログラミング教育（文部科学省、2017）は、主として技術・家庭科技術分野（以下技術分野）の中に内容「情報」として組み込まれている。技術分野における「情報」

のねらいは、「情報に関する基礎的・基本的な知識及び技術を習得させるとともに、情報に関する技術が社会や環境に果たす役割と影響について理解を深め、それらを適切に評価し活用する能力と態度を育成すること」としている。プログラミング教育によって、プログラム作成力の向上が目指されているのではなく、社会の中で活用される知識基盤の形成、および未知の問題に対してプログラミング的思考によって問題解決できる資質・能力が目指されている。2008年度に実施された学習指導要領によって、このようなねらいをもってプログラミング教育が必修化されたということは、技術分野では従来から、プログラミング教育によって、問題解決する力の育成が目指されていたのである。

これらの議論や指導事項を概観すると、いわゆる学校知と揶揄される学校教育における学習（稲垣ら、1998）を改善し、様々な局面において、

自ら問題を設定し、最適解を導き出せる力の育成が重視されていると考える。プログラミング教育においては、授業で提示した問題に対するプログラムを作成できればよいというものではなく、未知の問題に対してプログラムの思考をもとに解決策を導き出せる力の育成が求められているといえよう。

このような力の育成に関して、従前から技術分野では、「生活の課題に対して最適な解決策を追究することや生活を具体的に工夫することを体験することなどによって、生活をよりよくしようとする能力と態度をはぐくむことができる」(文部科学省, 2008)としている。「課題に対して最適な解決策を追求すること」は、課題に対する解が単一のものではなく、複数存在することを示しており、その上で、よりよい解決策を導き出せる学習こそが有効であると解釈できる。

プログラミングは、同じ動作でも様々な命令の組み合わせが存在する。プログラムが高速動作するという特性上、命令の組み合わせや手順に多少の無駄があったとしても、設定した問題の解決策として十分な動作を実現してしまうからである。プログラムの設計にあたっては、こうした特性を理解し、速度や汎用性を目的に、多様な組み合わせの中から、文脈に応じて最適化することが必要となる。このようなプログラム設計の意図を踏まえて、多様な命令の組み合わせから最適な組み合わせを探り出す活動が、プログラミング的思考による問題解決の力を育むと考えられる。ところが、先行的にプログラミング経験を持つ一部の中学生を除き、ほとんどの中学生がプログラミングをするための命令に関する知識・技能は、十分備わっていない。命令の持つ意味を理解できていないために、自らプログラムを構成したり、最適化したりできない。

先行実践事例(全日本中学校技術・家庭科研究会, 2016)を概観すると、技術分野のプログラミングの授業方法がいくつか報告されている。そこでは、問題提示とそれを解決する学習者のプログラム作成については報告されているものの、どのように学習者が命令を理解しているのかには言及されていない。おそらく、命令に関する知識を教師が伝達し、その上で生徒にプログラミングを促しているのではないかと考えられる。

このような授業は、学習者に、提示する問題

に対して命令のパラメータを変更するなどの工夫をさせて、文脈に沿った動作の最適化を考えさせることはできる。しかし、前述した多様な命令の組み合わせから最適化を考えさせる活動にはなりにくい。なぜなら、命令に関する知識を教師が与えているので、それ以上の認識の進展を妨げると考えられるからである(西林, 1997)。プログラムを最適化する力を育成するためには、まず、プログラムに多様性が存在することを理解し、それぞれのプログラムが、同じ動作であっても、命令の構成によって異なる意味を持つことを体験的に学習していくことが必要であろう。つまり、教師が命令やその組み合わせについて伝達するのではなく、学習者自身が動作を確認しつつ、命令や組み合わせ方の意味を獲得していく学習が期待される。

本稿ではこの点に着目し、学習者にプログラム構成の多様さに気づかせる教材、および授業方法を開発した。そして、開発した授業を実践し、プログラムの構成に多様性が表出することの確認を目的とした。

2. 研究方法

2.1 制御する教材

学習者にプログラムを構成させて、プログラムの命令やアルゴリズムの意味を理解させるためには、自ら構成したプログラムの動作を視覚的に確認することが有効と考えられる(岡本 et al., 2013)。このため、動作のON, OFFが簡単で、かつ視覚的に捉えられるLEDを用いることとした。動作を現実場面の文脈と比較させられることを考え、LEDを交差点の形状に配置し、押しボタンスイッチを取り付けることで、押しボタン式信号機の再現も可能になるようにし、ユニバーサル基板に作成した(図1)。

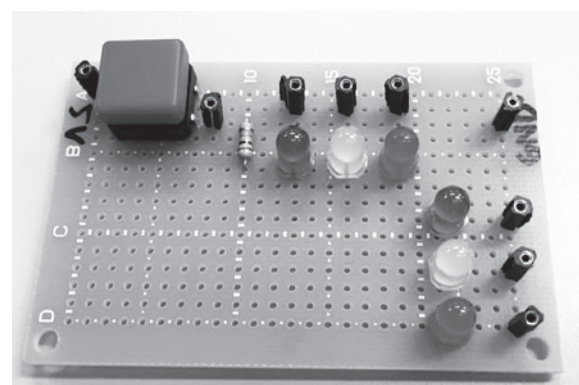


図1 作成したLED基板

また、このLED基板を用いてセンサー入力による自動化を可能とするため、光センサーや温度センサー等を取り付けたセンサー基板も作成した(図2)。

これらの基板やマイコンをブレッドボードジャンパーワイヤーで接続することで、光センサーや温度センサーの入力状況に応じて、LEDの点灯を制御させるようなテーマを提示できる。LEDを点灯、消灯させるだけの簡単なテーマから、交差点の光センサーと押しボタンによる複合制御のような、難易度の高いテーマの設定も可能であり、学習者の理解度に応じた、難易度設定ができる教材である。

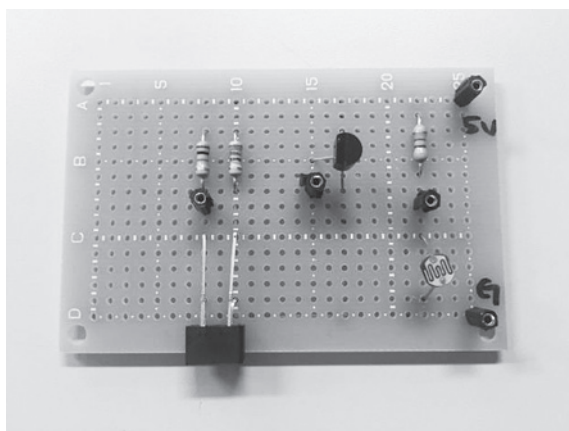


図2 自作センサー基板

教材の仕組みとしては、LEDを並列接続しているだけであり、また、押しボタン等も基板上に配置しているだけで、独立している。センサー基板についても同様で、電源と信号出力の端子をブレッドボードジャンパーワイヤーでLED基板やマイコンに接続できるようにした(図3)。

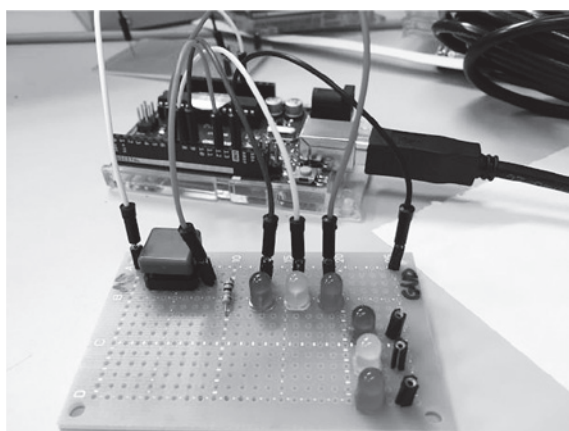


図3 ブレッドボードジャンパーワイヤーで接続

教材を単純な構成としたのは、接続をブラックボックス化するのではなく、ブレッドボードジャンパーワイヤーを用いて、自ら接続することで、信号の流れを確認しやすくするためである。接続は多少複雑化するものの、つながりが可視化されることで、プログラムで制御している実感を持たせることをねらった。

2.2 マイコンとプログラミングエディタ

プログラムを学習者が自ら構成できるためには、命令がどのような動作をするのか理解しておく必要がある。したがって、命令の動作について教師が伝達せずに学習者が試行錯誤の中から理解していくためには、命令の動作に関して、ある程度の推論を立てられるようにしておかなければならない。そのため、プログラミングに用いられるエディタが学習者にとって直感的であることが必要である。こうした要求にこたえるために、LEDの点灯や消灯を制御するマイコンとしてarduinoを扱い、プログラミングエディタとしてarduinoを動作させるプログラムを作成するsketchというエディタ、及び、そのプラグインであるardublockというビジュアルプログラミングエディタを用いることとした(図4)。

このardublockは命令をブロックでつなげてプログラミングでき、命令同士のつながりやパラメータ等の設定もブロックの形状を確認することで誤りなく構成できる。さらに、命令ブロックには日本語で動作が示されており、例えば、条件分岐の命令ブロックには「もし」と表示されている。

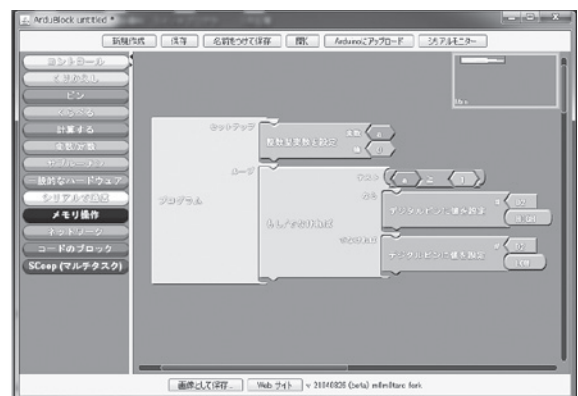


図4 授業で用いたプログラミングエディタ

2.3 カリキュラムおよび授業の流れ

ほとんどの学習者は、プログラミングに関する既有知識が十分ではなく、自らが思考してプログラムを構成することが困難である。そこで、比較的理解しやすく、生活経験との概念が近い処理から学習を始め、徐々に難易度が高くなるようなカリキュラムを設定した（表1）。

表1 学習内容の設定

学習する処理の内容	提示したテーマ
順次	LEDを点灯、消灯 LEDの点滅
ループ	LEDの点滅
サブルーチン	LEDを使ったモールス 信号
条件分岐	スイッチで動作する LED点灯

学習者に要求する処理について、順次から条件分岐へと、複雑な思考を必要とするものほど、後に学習場面を設定するようにしている。この処理の難易度設定をもとに、LED基板を点灯させるテーマを決定した

授業の流れは、3段階の構成とした（図5）。1段階目は、学習者にテーマを提示しただけで思考を促す段階である。教師が問題解決に向けての情報を何も語ることなく、学習者の思考が要求されることから、この段階でプログラミングが可能なのは、既有知識を有する学習者、もしくは、生活経験等を自らの思考によって転移させ、プログラムの内容に当てはめることのできた非常に勤が鋭い学習者と考えられる。この段階で学習者自身がプログラミングするのは非常に困難ではあるが、ブロックに表示された処理を理解しようとしたり、自分なりに組み合わせを考えて実行させた結果から処理の理解を進めたりできることから、あえてこの段階を設定した。この段階で、不確かながらも様々な命令と動作とのつながりを確認し、命令や処理の動作について理解することを期待した。

2段階目は、1段階目での試行錯誤を経て、教師が学習者に対して支援する段階である。この支援は、学習者に問題の解決方法を伝えるのではなく、問題解決に必要な処理に類似した動作を行わせ、そこから学習者に解決方法を発想

させるものである。1段階目で、様々な試行錯誤の中から処理とその動作について確認したことに加えて、この段階で動作を通じた手順の構成について思考することで、既存の動作イメージをプログラムの構成に転移させることをねらったものである。

3段階目は、ここまでの段階でプログラムを発想できない学習者に対して、その手段を伝達する段階である。1時間という授業の時間的限界があるため、全員が発想できるまで待つことができない。この段階まで発想できない学習者は、プログラミングの既有知識が不十分であったり、動作イメージから転移できなかったりするため、かなり詳細なイメージを理解させる必要がある。したがって、この段階までプログラムを作成できていない学習者には、具体的なプログラムを提示し、詳細に説明をする必要があると判断した。

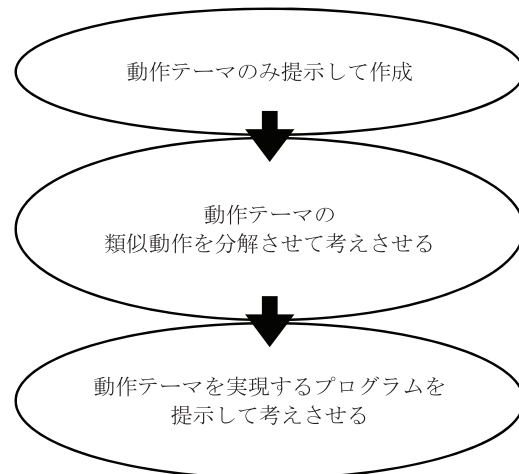


図5 授業の流れ

2.4 研究期間および被調査者

平成29年度4月から10月にかけて授業実践を行った。また、11月初旬頃に実技テストを行い、被調査者のプログラムに多様性が表出しているかを確認した。対象被調査者は鳥取大学附属中学校の3年生4クラス138人である。

実技テストは、全部で6問設定した（図6）。問題1から問題6にかけて、単にLEDを点灯させるだけの単純な問題から、ループ、サブルーチン、条件分岐等の各処理を複合的に必要とする問題へと徐々に難易度を高めた。被調査者には解答となるプログラムを作成したのち、指定するファイル名で、指定するフォルダに保存するよう指示した。

3年生実技テスト	
それぞれの問題を読み、プログラムを作成しなさい。 作成したプログラムは、指定したファイル名で、自分のフォルダに保存しなさい。 作成途中のプログラムも、破棄せず、保存しなさい。 arduinoと基板の接続は指定通りにしなさい。	
接続	D3, D6-赤 LED D4, D7-黄 LED D5, D8-青 LED D9-スイッチ 5V-5V GND-GND
問題 1	全ての LED (計 6 個) を点灯させる 指定ファイル名: mon1-○○△△ (○○は出席番号, △△は名前)
問題 2	D3 につないだ赤 LED を 0.5 秒間隔で、ずっと点滅させる。 指定ファイル名: mon2-○○△△
問題 3	D4 につないだ黄 LED を 0.5 秒間隔で、7 回点滅させ、最後は LED を消灯させる。 指定ファイル名: mon3-○○△△
問題 4	スイッチを押している間だけ、D5 につないだ青 LED が点灯する。 指定ファイル名: mon4-○○△△
問題 5	スイッチを押すと、D5 につないだ青 LED が 5 秒間点灯する。 指定ファイル名: mon5-○○△△
問題 6	始め D5 につないだ青 LED が点灯していて、スイッチを押すと、青 LED が消灯し、D4 につないだ黄 LED が 3 秒間点灯、その後黄 LED が消灯し、D3 につないだ赤 LED が 5 秒間点灯したのち、青 LED の点灯に戻る。 指定ファイル名: mon6-○○△△

図 6 被調査者に提示した実技テスト (抜粋)

この問題のうち、最もプログラムの多様性が表出すると考えられる問題 6 について、正解を導き出した被調査者のプログラムを確認し、分析した。問題 6 は、「始め D5 につないだ青 LED が点灯していて、スイッチを押すと、青 LED が消灯し、D4 につないだ黄 LED が 3 秒間点灯、その後黄 LED が消灯し、D3 につないだ赤 LED が 5 秒間点灯したのち、青 LED の点灯に戻る」という動作をプログラムによって実現する問題である。D3,D4,D5 というのは、マイコンの出力ピンのことを指しており、それぞれ LED 基板の赤、黄、青につなげるように指示してある。待機状態で青 LED 点灯操作、スイッチにより作動する赤、黄、青 LED の点灯、消灯操作を要求する問題となっている。

この問題で多様性が表出すると考える理由は、ループと条件分岐を組み合わせるプログラムを構成する必要があること。高速動作するというプログラムの特徴、およびループの特徴を利用してプログラミングしなければならないこと。条件分岐の手段に選択の余地が多いこと。サブルーチンで構成することもできることである。

3. 結果

3.1 表出したプログラムについて

実技テストの結果において、どのようなプロ

グラムが表出しているのかを確認した。その結果、同じ動作であっても、様々な構成のプログラムが表出していることが明らかになった。これらの表出したプログラムを大きく 4 種類に分類した。

3.2 「もし／でなければ」で構成したプログラム

問題として提示した言葉の文脈を、2 つに動作にわかれるプログラムと捉えた時に発想されるプログラムであろうと考えられる。「もし／でなければ」という命令を用いて、スイッチが押されていない時に青 LED を点灯させ、押された時に黄、赤 LED を点灯させるように実行内容を変えている (図 7)。



図 7 「もし／でなければ」で構成したプログラム例

この「もし／でなければ」という命令は、一般的には「if ~ else ~」という命令である。二つ以上に分岐させる際に用いられる命令であるため、問題の文脈からは、最も考えやすい処理の流れと考えられる。

この処理が単独であれば、一度分岐して終了してしまうが、条件分岐命令がループに組み込まれているため、高速で何度も判断が行われる。つまり、スイッチのオンとオフが高速で判断され続けている状態であり、スイッチが押されているのか、押されていないのか常に監視しているプログラムである。

3.3 「もし」を並べて構成したプログラム

問題の文脈を 2 つの条件と捉えたときに表出すると考えられるプログラムである。「もし」という命令を 2 つ用いて、2 回判断の場面を設定したものである。例えば、最初にスイッチが押されていないことを判断させ、押されてい

ければ青LEDを点灯させる。次に、スイッチが押されているかを判断させ、押されていれば黄、赤LEDの点灯を実行させる(図8)。

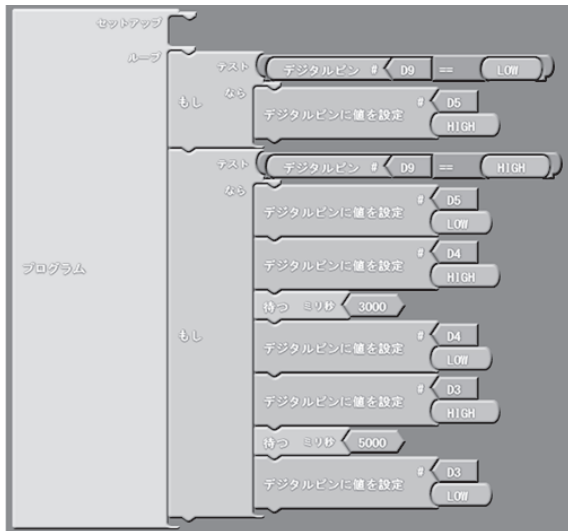


図8 「もし」を並べて構成したプログラム例

この「もし」という命令は、一般的には「if～」という命令である。条件に該当した時のみ指定する処理を実行する命令である。この処理の特徴は、この命令が実行された瞬間に判断を実行し、条件に該当しないときは、何も処理を行わず、すぐにプログラムが流れてしまうことである。したがって「もし」という命令を2つ並べて実行しても、一瞬で処理が流れてしまう。このため、問題6のようなスイッチ動作させるプログラムでは扱いにくいとも考えられる。しかし、これもループの中で動作させることによって高速で判断を繰り返すことになり、ボタンを押した瞬間にどちらかの条件に当てはまるようにできるプログラムである。

3.4 「もし」をひとつだけ用いたプログラム

「スイッチを押していない」という条件は、「何も操作していない」と同じであると捉え、条件はひとつと考え、このプログラムが発想される(図9)。

まず青LEDを点灯させた上で、スイッチが押された時のみ、黄、赤LEDを点灯させるようにしたものである。これもループの中で動作させることで、高速で何度も条件を判断させているため、スイッチを押した瞬間に分岐して黄、赤LEDの点灯動作を実行させる。

条件分岐を2つ並べるプログラムよりも簡略

化されている。

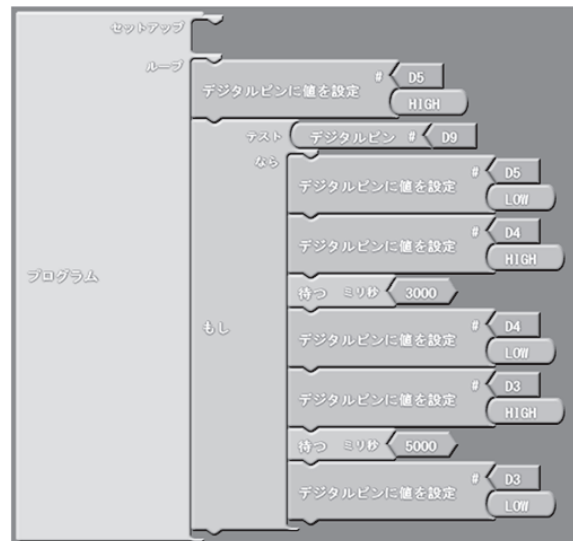


図9 「もし」をひとつ用いたプログラム例

3.5 サブルーチンを用いたプログラム

プログラムの構造としては、「もし」をひとつだけ用いたプログラムと同じである。しかし、スイッチを押した時に条件分岐した動作部分をサブルーチン化している(図10)。

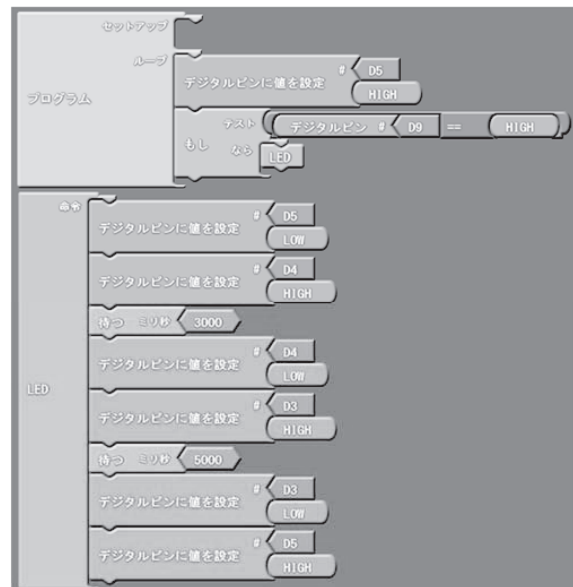


図10 サブルーチンを用いたプログラム例

このことで、プログラムの構成自体には、「もし」をひとつだけ用いたプログラムと大きな違いがないものの、視覚的に構造化されたプログラムになっている。黄、赤LEDの点灯動作をひとつの動作の塊として捉えてサブルーチン化できるということは、条件分岐部分と条件によ

る黄、赤 LED の点灯操作部分とを別構造として考えており、それを視覚的に構成したプログラムといえる。このようなプログラムが作成できるということは、スイッチ入力に対応する動作部分を、後に変更したり、黄、赤 LED の動作を他の場面で用いたりする可能性を考慮しているといえる。プログラムの汎用性について思考が及んでおり、それぞれの命令をどのように構成するのか、プログラム自体の理解が相当深まっていると考えられる。

4. 考察

授業実践より、LED 点灯を制御する教材を用い、試行錯誤を支援することで学習者が実践的にプログラムの構造を理解してプログラミングできることがわかった。その結果、プログラミングに多様性が表出していることを確認した。

学習者が作成したプログラムの構成が多様なものとなったのは、始めから正解とされるプログラムを提示せず、学習者が自分の思考をもとにプログラミングしたことが要因として挙げられる。正解が与えられなかったからこそ、目指す動作を構成する命令、及びその手順や組み合わせを自分で思考しなければならない必然性が生じ、学習者を試行錯誤によるプログラミングへと働きかけた。このように試行錯誤をもとにしたプログラミングを可能にしたことについて、いくつか考察した。

まず、学習者は提示された問題の文章に着目し、文章を分解して命令の手がかりを導き出したと推察する。多様な表出を見せたプログラミングにおいて、その命令の組み合わせ方をみると、文章の表現を学習者がどのように噛み砕いたのかによって、構成が異なっていると考えられる。つまり、提示した文章から学習者それぞれが各自の条件分岐の捉え方を想起し、それが条件分岐を実現させる命令の組み合わせ方に対して直接的に作用している。「もし／でなければ」や「もし」をふたつ用いたプログラムは、条件分岐処理を細かく分解した時の各自のイメージが表出したものと考えられる。固定した条件分岐のプログラムを伝達した場合に、このような複数の条件分岐イメージは表出されないと考えられる。

次に、LED 点灯教材の実行結果は同じであるが、そこに至る命令の処理機能について理解する過程が多様であったと考えられる。3 段階

の授業構成の中で、第 1 段階では、手探りで命令の意味を確認していくことが要求される。とりあえず命令を並べて実行した結果、どのような動作をするのかを学習者は教材の観察によって確認する。そして、プログラムの動作が高速で実行されるという特徴は、「もし／でなければ」と「もし」の持つ機能の差異を明確に伝えてくれる。自ら命令を選択してプログラムを構成し、実行結果を確認する行為によって、命令がどのような動作になるのかを理解していくのである。このように、教材を通した視覚情報と自分の頭の中で描いた命令の構成情報とを比較し、命令の処理機能について理解を深めていくと考えられる。

次に、支援として導入した動作の分解によって、手順の構成を発想させ、学習者の思考によるプログラミングを可能にしたと考えられる。提示された文章によって LED の動作イメージは想起されるものの、細かい手順に変換することに関しては、中学生段階の学習者にとっては困難と考えられる。動作の分解という支援は、動作イメージを細かい命令の分解と捉えさせ、LED の動作イメージも同様に細かい命令の分解と捉えさせる。その結果、分解されたイメージをもとに命令を構成して行き、学習者のイメージに即したプログラミングを可能とする。

最後に、授業実践で用いた LED 点灯教材とシングルボードコンピュータ及び、プログラミングエディタが、学習者の思考難易度を下げ、プログラミングの理解に大きな影響をもたらしたと考えられる。LED を教材としたことで、プログラムの動作結果が光に対応した速度で捉えられるようになった。このことが、プログラムの処理が高速動作していることを捉えやすくしていると推察する。また、操作しているコンピュータにつないだマイコンを経由してプログラムの動作をさせていることで、プログラムによる動作がコンピュータ内のヴァーチャルなものではない現実的な感覚を与えてくれると考える。さらに、プログラミングエディタがテキスト入力による言語表現を不要としていることで、動作の感覚的なイメージをプログラミングすることを可能にしたと考える。

シングルボードコンピュータを PC に接続してプログラミング学習を行ったところ、以上に考察したように、様々な効果が得られた。単に授業者に伝達されたプログラムを模写すること

にとどまらない、学習者が主体的に思考してプログラム学習に取り組む授業が作られていたものとする。

5. 今後の課題

このような多様なプログラミングを表出させたのは、ただ教材を準備したからではなく、授業展開に思考場面を多く設定し、授業者が学習者に対して適切に支援をしたこと大きな要因と考えられる。教材と授業者の授業内での支援、そして、学習者に思考させる時間的な保証が、より深い思考へと学習者を誘うと考えられる。今後は、このような点を踏まえて、授業者の支援を念頭において教材を改良していきたいと考えている。また、プログラミング時における思考プロセスに関しては、不明確な部分が多い。このプロセスの研究が進めば、より効果的なプログラミング学習が望めると考えられることから、プログラミング時の思考プロセスを解明することも視野に入れて研究を進めていきたい。

6. 参考文献

稲垣佳世子, 羽多野諠余夫 (1998) '学校化された学びのゆがみ', in 岩波講座 3 現代の教育 授業と学習の転換, pp. 70-91.

岡本雅子 et al. (2013) 「視覚的顕在化」に着目したプログラミング学習教材の開発と評価', 日本教育工学会論文誌, 37 (1), p. 35 - 45.

西林克彦 (1997) 「わかる」のしくみ—「わかったつもり」からの脱却. 新曜社.

全日本中学校技術・家庭科研究会 (2016) 中学校技術・家庭科理論と実践 No.54.

文部科学省 (2008) 中学校学習指導要領解説 技術・家庭編. 文部科学省.

文部科学省 (2016) 次期学習指導要領等に向けたこれまでの審議のまとめ (第2部).

Available at: http://www.mext.go.jp/component/b_menu/shingi/toushin/__icsFiles/afieldfile/2016/09/09/1377021_1_2.pdf.

文部科学省 (2017) 中学校学習指導要領解説 技術・家庭編. 文部科学省.