

LLMを用いたラダープログラムの処理内容説明の生成技術

LLM-Based Program Analysis for Ladder Logic

*情報技術総合研究所

要旨

ラダー言語はPLC(Programmable Logic Controller)で制御する内容を記述するプログラミング言語であり、様々な設備の制御に使用されている。制御設計者が設備を改修する際にはラダー言語で記述された制御プログラム(以下“ラダープログラム”という。)を解読して処理内容を理解する必要があるが、この作業は負担が大きい。

そこで三菱電機は、LLM(Large Language Models)を用いてラダープログラムを解析し、処理内容の説明を生成する技術を開発した。ラダープログラムだけでなくPLCのログデータも合わせて解析し、信号の変化順にラダープログラムの処理内容を説明することが特長である。これによって設備内の機器の制御順序が分かりやすい説明になり、設計者のラダープログラム解読の負担を軽減できる。

1. まえがき

PLCは様々な設備の制御に使用されている。PLCが設備の制御を行う際には、設備内の機器から信号を受け取って、信号を制御プログラムで処理し、設備内の機器に対して信号を出力する、ということを繰り返す。制御プログラムを記述するプログラミング言語は幾つか存在するが、日本ではラダー言語というプログラミング言語が特に多く使用されている。

図1を用いて、ラダープログラムによる設備制御の概要を述べる。図1(a)は、PLCと機器(スタートスイッチ、プッシャー)で構成された設備を表しており、図中のX0、Y10などはPLCと機器の間でやり取りされる入出力信号である。このようなPLCと機器で構成された設備では、PLCは機器から入力信号を受け取って、その信号をラダープログラムで処理して出力信号を機器に渡すことによって、機器の制御を行う。

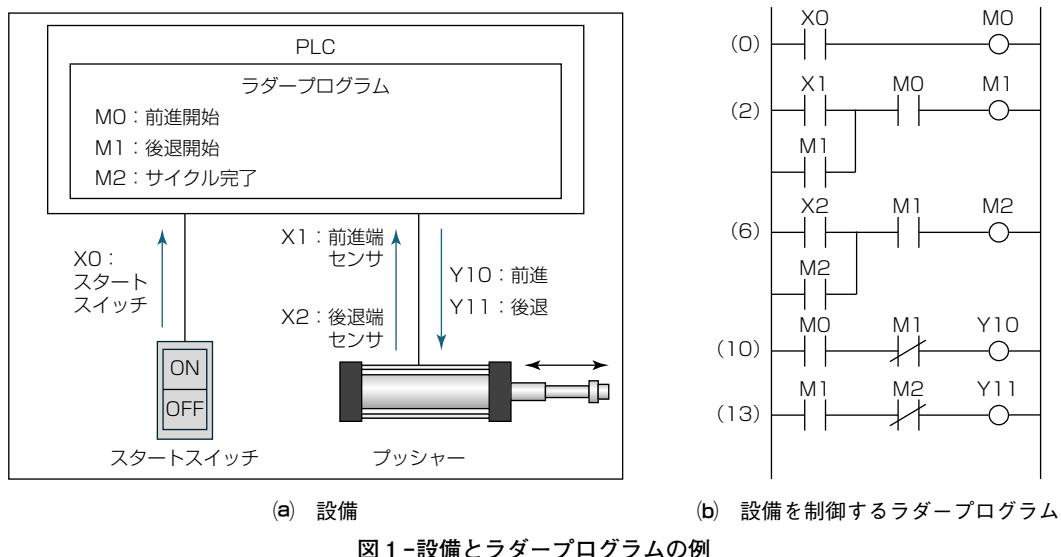


図1-設備とラダープログラムの例

図1(b)は、図1(a)の設備を制御するラダープログラムの例である。ラダープログラムでは左右の縦線の間に横線を引いて、横線の左側に条件、右側に処理を記述して、どのような場合にどのような処理を行うかを表す。例えば、図1(b)の(0)について、左の部分は“X0が1のとき”という条件、右の部分は“M0を1にする”という処理を示している。同様に、図1(b)の(10)では、左の部分で“M0が1、かつM1が0のとき”という条件、右の部分は“Y10を1にする”という処理を示して

いる。図1(b)の(0), (10)の条件、処理を組み合わせると“X0が1、かつM1が0のときY10を1にする”という制御になる。この制御によって、設備のスタートスイッチをONにすると、プッシャーは前進端に到達するまで前進する。

このように、ラダープログラムは複数の入力信号の値やその変化順序が処理に影響し、かつ一つの出力信号に関係する処理がラダープログラムの複数箇所に分けて書かれているため、解読が難しい。

2. ラダープログラムの制御内容理解の課題

制御設計者が設備を改修する場合や、設備の不具合の原因を推定する場合には、設備がラダープログラムによってどのように制御されているのかを理解している必要がある。しかし、長く使われている設備では、当時設計を行った制御設計者が退職などで在籍していない場合があり、ラダープログラムの解読が必要になる。

図2は、図1(b)のラダープログラムを人が解読した結果を文章で表した例である。この文章では、装置の信号がどのような順序で変化するか、信号変化がラダープログラムのどの部分によって発生するかが分かるようになっている。文中のステップは図1(b)の左に書かれた番号と対応している。このような処理内容の説明があれば、制御設計者が設備の改修を行う場合や、設備の不具合の原因を推定する場合に、ラダープログラムのどの部分を確認、修正すればよいか判断できる。

しかし、1章で述べたように、ラダープログラムの解読は難しく、制御設計者が大規模なラダープログラムを解読して図2のような説明を作る作業は負担が大きい。

そこで当社では、LLMを用いてラダープログラムの処理内容説明を生成する技術を開発した。この技術で生成したラダープログラムの処理内容説明は、設備内の機器をどのような順序で制御しているのか分かりやすいことが特長である。

ユーザーがスタートスイッチをONにするとX0が1に変化し、プログラムの0ステップ目の処理によりM0(前進開始)が1に変化する。M0(前進開始)が1に変化するとプログラムの10ステップ目の処理によりY10(前進)が1に変化し、プッシャーが前進を開始する。

プッシャーが前進端に到達するとX1(前進端センサ)が1に変化し、プログラムの2ステップ目の処理によりM1(後退開始)が1に変化する。M1(後退開始)が1に変化するとプログラムの13ステップ目の処理によりY11(後退)が1に変化し、プッシャーが後退を開始する。また、プログラムの10ステップ目の処理によりY10(前進)が0に変化するため、プッシャーの前進は停止する。

プッシャーが後退端に到達するとX2(後退端センサ)が1に変化し、プログラムの6ステップ目の処理によりM2(サイクル完了)が1に変化する。M2(サイクル完了)が1に変化するとプログラムの13ステップ目の処理によりY11(後退)が0に変化し、プッシャーの後退が停止する。

ユーザーがスタートスイッチをOFFにするとX0が0に変化し、プログラムの0ステップ目の処理によりM0(前進開始)が0に変化する。M0(前進開始)が0に変化するとプログラムの2ステップ目の処理によりM1(後退開始)が0に変化する。M1(後退開始)が0に変化するとプログラムの6ステップ目の処理によりM2(サイクル完了)が0に変化する。

図2-ラダープログラムの処理内容の人による解説結果例

3. ラダープログラムの処理内容説明の生成方法

LLMを用いてプログラムの説明を生成する場合、基本的にラダープログラムの上から順に説明が生成される。そのため、ラダープログラムだけを入力データとしてLLMで処理内容説明を生成するだけでは、図2に示したような、ラダープログラムで行われる処理の順序が分かる説明が生成できない。この技術ではラダープログラムだけでなくPLCのログデータを入力データとして用いることで、この課題を解決する。ラダープログラムの処理内容説明を生成するまでの手順と、各手順での入力データ、出力データを、表1に示す。

表1-この技術での処理内容説明の生成手順

No.	手順	入力データ	出力データ	説明
1	ラダープログラムの分割	ラダープログラム	分割したラダープログラム	3.1節参照
2	信号変化順の抽出	ログデータ	各時刻での信号変化の情報	3.2節参照
3	信号変化とラダープログラムの対応付け	No.1とNo.2の出力データ	信号変化とラダープログラムの関係情報	3.3節参照
4	処理内容説明の生成	No.3の出力データと、各信号の意味	ラダープログラムの処理内容説明	3.4節参照

3.1 ラダープログラムの分割

図2の解読結果では、最初にラダープログラムの0ステップ目の処理について説明し、その次にラダープログラムの10ステップ目の処理が行われることを説明している。ラダープログラムでは条件を満たしている部分だけ処理が行われるため、このようにプログラムの処理の並び順と実際に処理が実行される順序が一致しない場合がある。

そこで、この技術では並び替えを可能にするために、図3のように、ラダープログラムの分割を行う。分割は、処理の内容と、その処理を行う条件の組合せである回路ブロック単位で行う。そして、後の手順で正しい順番での並び替えを可能にするために、条件部分と処理部分についても分割する。

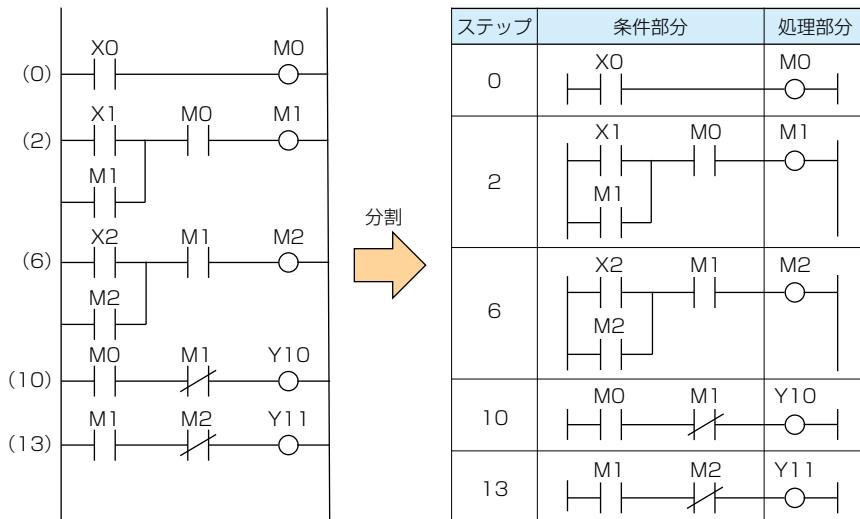


図3-ラダープログラムの分割イメージ

3.2 信号変化順の抽出

図2の解読結果では、最初にX0が1になり、その後X1が1になるなど、入力信号の変化順が示されている。このような入力信号の変化は機器の動きによって起こるものであり、ラダープログラムからは入力信号がいつどのように変化するかを読み取ることができない。

そこで、この技術では、ラダープログラムに加えて、PLCのログデータを入力データとして使用する。PLCのログデータは設備動作中の各信号の状態を時刻とともに記録したもので、図4のように、表形式で表すことができる。例えば図4では、時刻列が00:01.0の行でX0列の値が1であることから、00:01.0の時刻でX0の値が1であることが読み取れる。

この技術では、図4のように、各信号の値が前の時刻から変化している部分をログデータから抽出することで、処理内容の説明の生成に必要になる、信号がいつどのように変化したのかを表す信号変化順の情報を作成する。

前時刻から値が変化している部分								
時刻	X0	X1	X2	Y10	Y11	M0	M1	M2
00:00.0	0	0	1	0	0	0	0	0
00:01.0	1	0	1	1	0	1	0	0
00:02.0	1	0	0	1	0	1	0	0
00:03.0	1	0	0	1	0	1	0	0
00:04.0	1	1	0	0	1	1	1	0
00:05.0	1	0	0	1	1	1	1	0
00:06.0	1	0	1	0	0	1	1	1
00:07.0	0	0	1	0	0	0	0	0

時刻	変化
00:01.0	X0が1に変化 Y10が1に変化 M0が1に変化
00:02.0	X2が0に変化
00:04.0	X1が1に変化 Y10が0に変化 Y11が1に変化 M1が1に変化
00:05.0	X1が0に変化
00:06.0	X2が1に変化 Y11が0に変化 M2が1に変化
00:07.0	X0が0に変化 M0が0に変化 M1が0に変化 M2が0に変化

図4-ログデータからの変化抽出イメージ

3.3 信号変化とラダープログラムの対応付け

図2の解読結果では、信号変化とラダープログラムの該当部分を対応付けて説明している。このような説明を生成するためには、3.1節の処理でラダープログラムの分割を行った結果と、3.2節の処理で信号変化的抽出を行った結果の対応付けを行う必要がある。

この技術では、対応付けの候補を抽出し、候補になった対応付けが正しいかをLLMを用いた検証によって確認する。

対応付けの候補を見つける手順では、3.2節の処理で抽出した信号変化を基に3.1節の処理で分割したラダープログラムから、条件部分の信号と、処理部分の信号の両方が同時刻で変化しているプログラムを抽出する。図5の例では、00:01.0の時刻では三つの信号変化が発生しているが、条件部分と処理部分の両方にこれらの変化が含まれるラダープログラムは、ステップ0とステップ10のラダープログラムである。この二つのプログラムで、条件となる信号の値が変化することで処理が行われ、他の信号の値が変化した可能性があると予測できる。

候補になった対応付けが正しいかをLLMを用いて検証する手順では、信号の値とラダープログラムをLLMに与えて、ラダープログラムの処理で信号の値が変化するかを確認する。この手順が必要である理由は、先に述べた、抽出した対応付けの候補が正しいとは限らないためである。例えば、図5のステップ10のプログラムは条件部分にM1の信号が含まれており、M1の信号の値次第では、M0が1になってもY10が1にならない。また、同じ信号の値を変化させる処理の候補がラダープログラム内に複数存在する場合には、実際に信号を変化させた部分がどの候補かを、この方法で確認する必要がある。

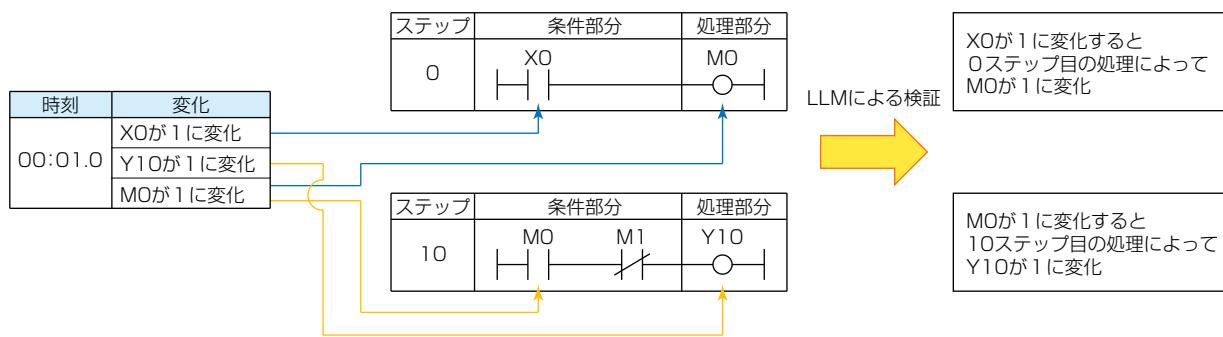


図5-信号変化とラダープログラムの対応付けイメージ

3.4 処理内容説明の生成

最後に、3.3節で得られた説明を結合して、ラダープログラムの処理内容説明を生成する。このとき、“X0(スタートスイッチ)”のように、信号だけでなく各信号の意味も説明に加えることで、設備の動作との対応を分かりやすくしている。

4. 検 証

この技術によって、ラダープログラムの処理内容を正しく記述した説明が生成できること、及び信号の変化順に処理内容を説明した分かりやすい説明が生成できることを確認するために、検証を行った。検証ではLLMとしてgpt-4o(2024-05-13)を使用した。

4.1 処理内容を正しく記述した説明が生成できることの検証

この技術で図1(b)のラダープログラムの処理内容説明を生成し、その結果を図2の人による解読結果と比較することで、処理内容を正しく記述した説明が生成できているかを検証した。図6は、この技術によるラダープログラムの処理内容説明の生成結果である。図2と図6を比較すると、“プッシャーが前進端に到達”のような信号変化の原因は入力データに情報がないため図6では記載されないが、“前進”“前進端センサ”的な各信号の意味からおおよその動作の予測は可能である。それ以外は図2と図6で同等の内容が記述されており、処理内容を正しく記述した説明が生成されている。

X0(スタートスイッチ)が1に変化すると、プログラムの0ステップ目の処理によりM0(前進開始)が1に変化します。
M0(前進開始)が1に変化すると、プログラムの10ステップ目の処理によりY10(前進)が1に変化します。

3.0秒後X1(前進端センサ)が1に変化すると、プログラムの2ステップ目の処理によりM1(後退開始)が1に変化します。
M1(後退開始)が1に変化すると、プログラムの13ステップ目の処理によりY11(後退)が1に変化します。
さらにプログラムの10ステップ目の処理によりY10(前進)が0に変化します。

2.0秒後X2(後退端センサ)が1に変化すると、プログラムの6ステップ目の処理によりM2(サイクル完了)が1に変化します。
M2(サイクル完了)が1に変化すると、プログラムの13ステップ目の処理によりY11(後退)が0に変化します。

1.0秒後X0(スタートスイッチ)が0に変化すると、プログラムの0ステップ目の処理によりM0(前進開始)が0に変化します。
M0(前進開始)が0に変化すると、プログラムの2ステップ目の処理によりM1(後退開始)が0に変化します。
M1(後退開始)が0に変化すると、プログラムの6ステップ目の処理によりM2(サイクル完了)が0に変化します。

図6-この技術によるラダープログラム処理内容説明の生成結果

4.2 信号の変化順に処理内容の説明を生成できることの検証

この技術を用いて生成した処理内容説明と、ラダープログラムだけを入力データとして生成した処理内容説明を比較し、3章で述べた手順の効果を確認した。ラダープログラムだけを入力データとして生成した処理内容説明を図7に示す。

図7に示した説明はラダープログラムの上から順に説明を作成しているため、プログラムのどの部分がどの順序で処理を行うかが判断できない。また、各信号の変化がどの順序で発生するかも分からず。一方、図6に示すように、この技術を用いて生成した処理内容説明ではこれらの問題を解決し、ログデータでの信号の変化順にラダープログラムの処理内容を説明できている。このような説明があれば、ラダープログラムの解読にかかる設計者の負担を軽減可能である。

1. X0がONのとき、M0をONにします。
2. X1がON、またはM1がON、かつM0がONのとき、M1をONにします。
3. X2がON、またはM2がON、かつM1がONのとき、M2をONにします。
4. M0がON、かつM1がOFFのとき、Y10をONにします。
5. M1がON、かつM2がOFFのとき、Y11をONにします。

図7-ラダープログラムだけを入力データとした処理内容説明の生成結果

5. むすび

LLMを活用し、ログデータから抽出した信号変化とラダープログラムを対応付けることによって、設備内の機器の制御順序が分かりやすい説明を生成する技術について述べた。この技術を活用することによって、設計者のラダープログラム解読の負担を軽減できる。

今後は処理内容についての質問に回答するチャットボットや仕様との差異の説明機能など、この技術を更に発展させた技術についても検討する。