2025.8.8～9
全国高等学校情報教育研究会
第18回全国大会(千葉大会)

生成AIによるプログラミング支援システムの開発とマイコンボードへの活用

天良(てんら)和男

元東京都立日比谷高等学校
元東京学芸大学

プログラミングでの生成AIの利用の背景と現状

背景

- 近年，生成AIの進化によりプログラムコードの自動生成が可能になる
- 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

現状

- 生成AIが出力したコードの実行がPC上に留まっている
- 生成AIに抽象的なプロンプト（指示文）を入力しても正しいコードが出力されることがあり，安易にAIに依存し，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

生成AIによるプログラミング支援システムの開発

- 独自に開発したクラスのメソッド
  生成AIが出力したコードをPC上で実行するだけでなく，マイコンボードに直接転送して実行し，ロボットや鉄道模型などの動作を制御
  ➡ 生成コードを手動で実行環境にコピー＆ペーストせずに，マイコンボードへ自動的に直接転送・実行可能
- 独自開発した出力調整プロンプト
  具体性に欠ける抽象的なプロンプト（指示文）では，適切なコードが生成されないようにした

出力調整プロンプトとは

- 生成AIを利用する際の問題点
  - 例：「二分探索のプログラムを作成して提出しなさい」
  - 生成AIに，「二分探索」と入力するだけで，即座にコードが生成されてしまう
  - 学習者が生成コードを自分で考えたものだと偽って提出する可能性がある
- 本システムでの解決策
  - 「真ん中から半分に分けて条件に合わない方を消していく」といった具体的なアルゴリズムをプロンプトに入力しない限り，適切なコードが生成されないようにする仕組み「出力調整プロンプト」を考えた
  - これにより，安易にAIに依存し，思考力や試行錯誤する力の低下を防ぐ

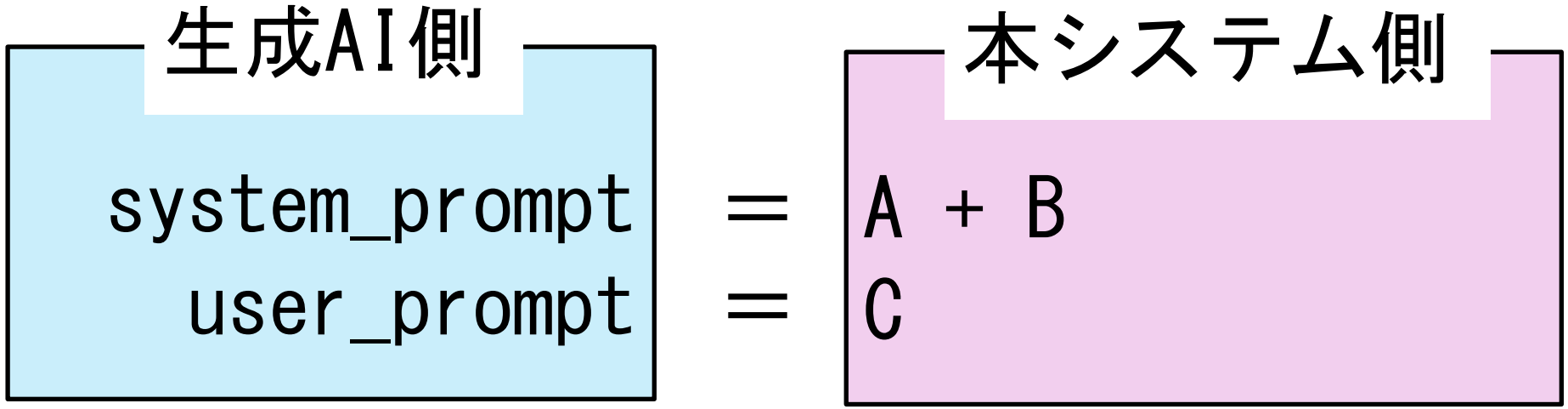
プロンプト（本システム側）

- ハードウェアプロンプト
  マイコンボードの入出力ポートや接続されるセンサやモータ，LEDの仕様など，ハードウェアに関する情報
- 出力調整プロンプト
  具体性に欠ける抽象的なプロンプトでは，適切なコードが生成されないようにする仕組み
- ユーザ作成プロンプト
  学習者が生成AIに対して入力する指示であり，ロボットの動作などのアルゴリズム

プロンプト（生成AI側）

- system_prompt
  システム（AI）の振る舞いや話し方，制限事項（生成AIが避けるべき応答や提供してはならない情報）を設定する
- user_prompt
  ユーザがAIに対して入力する具体的な指示や質問

プロンプト構成



A：ハードウェアプロンプト
B：出力調整プロンプト
C：ユーザ作成プロンプト

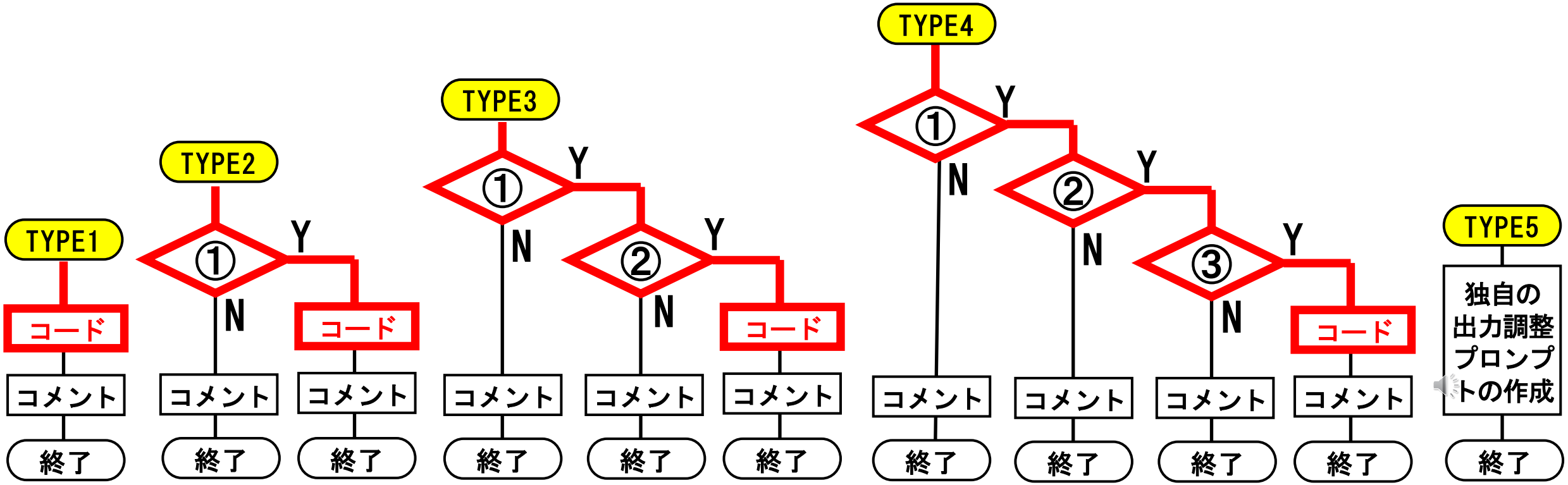
出力調整プロンプトのタイプ

- ５つのタイプ
- 学習者のスキルレベルに応じたコードや
  コメントを生成AIから出力できるように調整

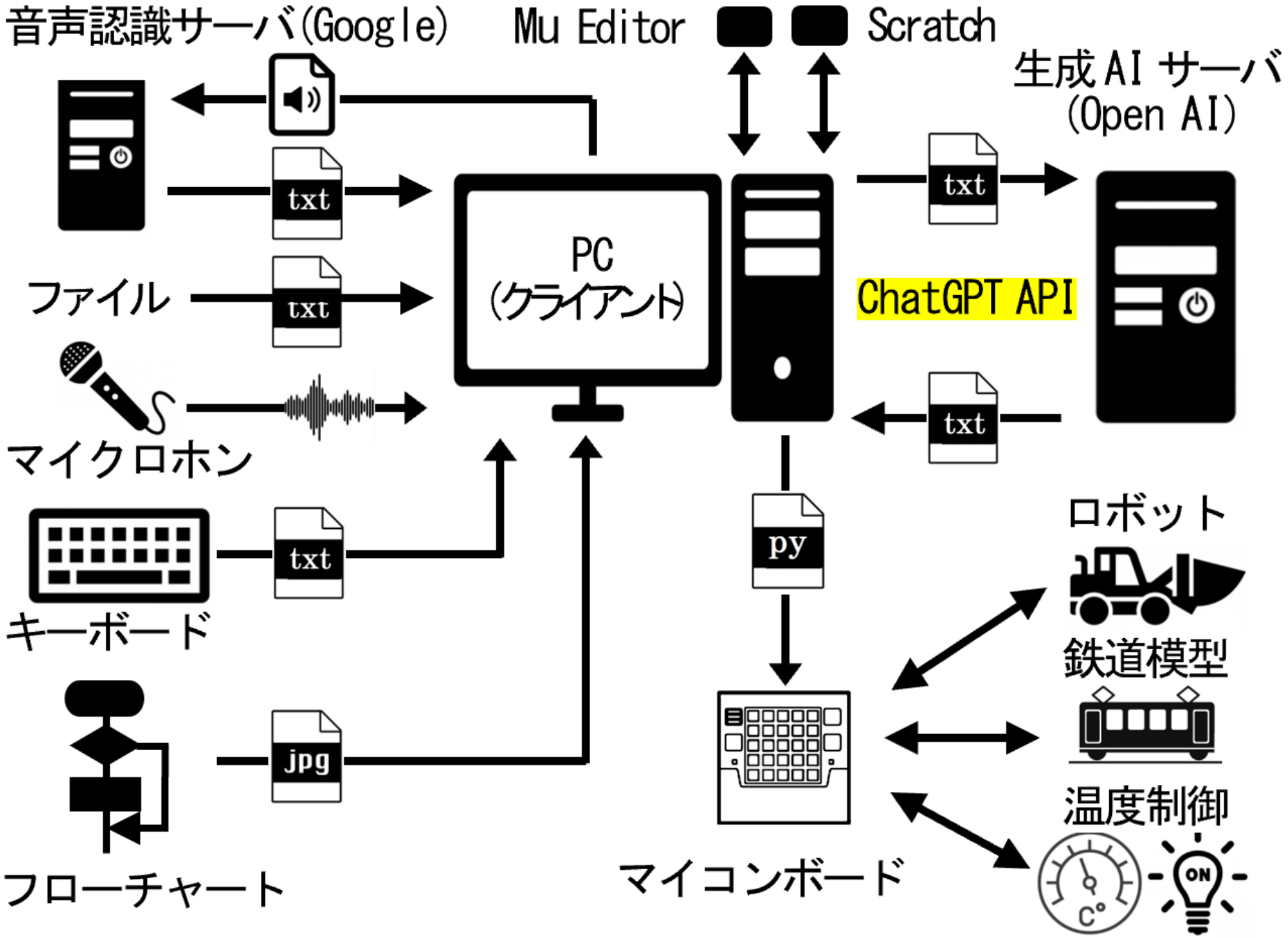
タイプ	概要
TYPE1	初級：無条件コード出力，詳細コメント
TYPE2	中級：条件付きコード出力，簡略コメント
TYPE3	上級：条件付きコード出力，簡略コメント
TYPE4	特上級：条件付きコード出力，簡略コメント
TYPE5	追加：独自の出力調整プロンプト

出力調整プロンプトのタイプ

ユーザ作成プロンプトの記述の中に①～③の記述があるか
①センサやモータ，LEDなどの部品の使用の記述
②センサやモータ，LEDなどの部品の接続情報の記述.
③センサやモータ，LEDなどの部品を制御するメソッドの使用の記述



システム構成図

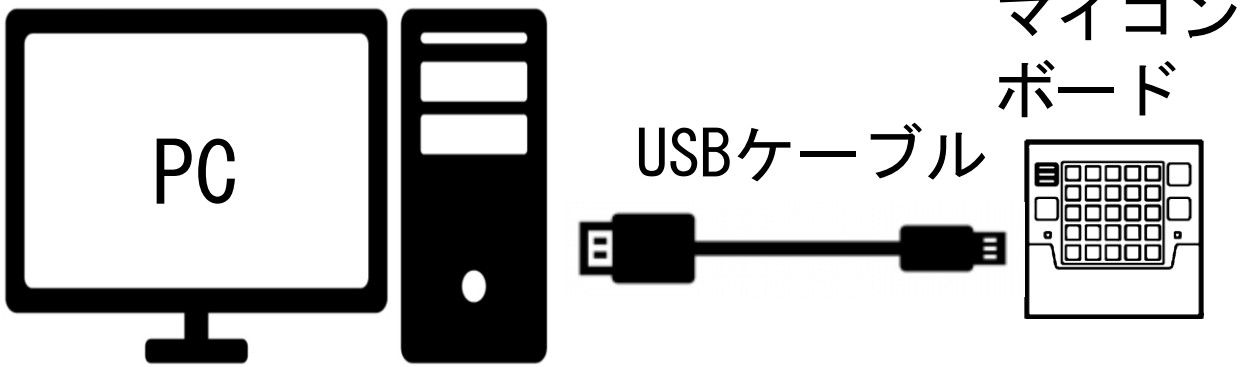

音声認識サーバ(Google)
Mu Editor
Scratch
生成AI サーバ
(Open AI)
txt
PC
(クライアント)
ファイル
ChatGPT API
マイクロホン
py
ロボット
キーボード
鉄道模型
jpg
温度制御
フローチャート
マイコンボード
ON

生成コードをマイコンボードへ転送・実行する仕組み

独自開発のクラスのメソッド

- MicroPythonコード(文字列)をUTF-8形式のバイナリーデータに変換
- シリアル通信を使用してマイコンボードに転送
- Raw REPLにより，非対話で全行一括実行

シリアル通信ではバイナリーデータで送る必要があるため，文字列をUTF-8形式のバイナリーデータに変換する



開発したソフトウェア（Intelligence Software）

❶ ハードウェアプロンプト

❷ 出力調整プロンプト

❸ ユーザ作成プロンプトの入力部分

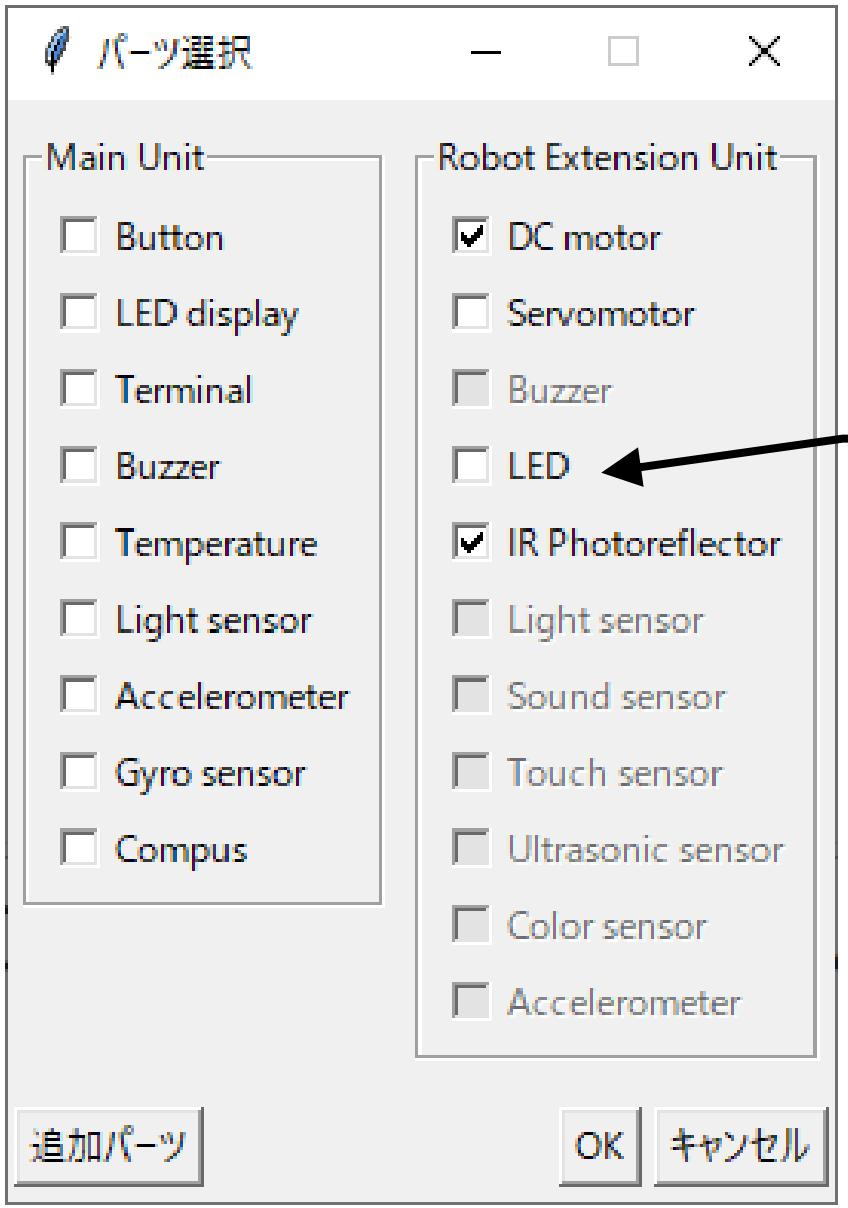
❹ 音声，画像，ファイルの入力

❺ 生成AIへの質問開始

❻ 生成AIから出力されたコードやコメント表示部分

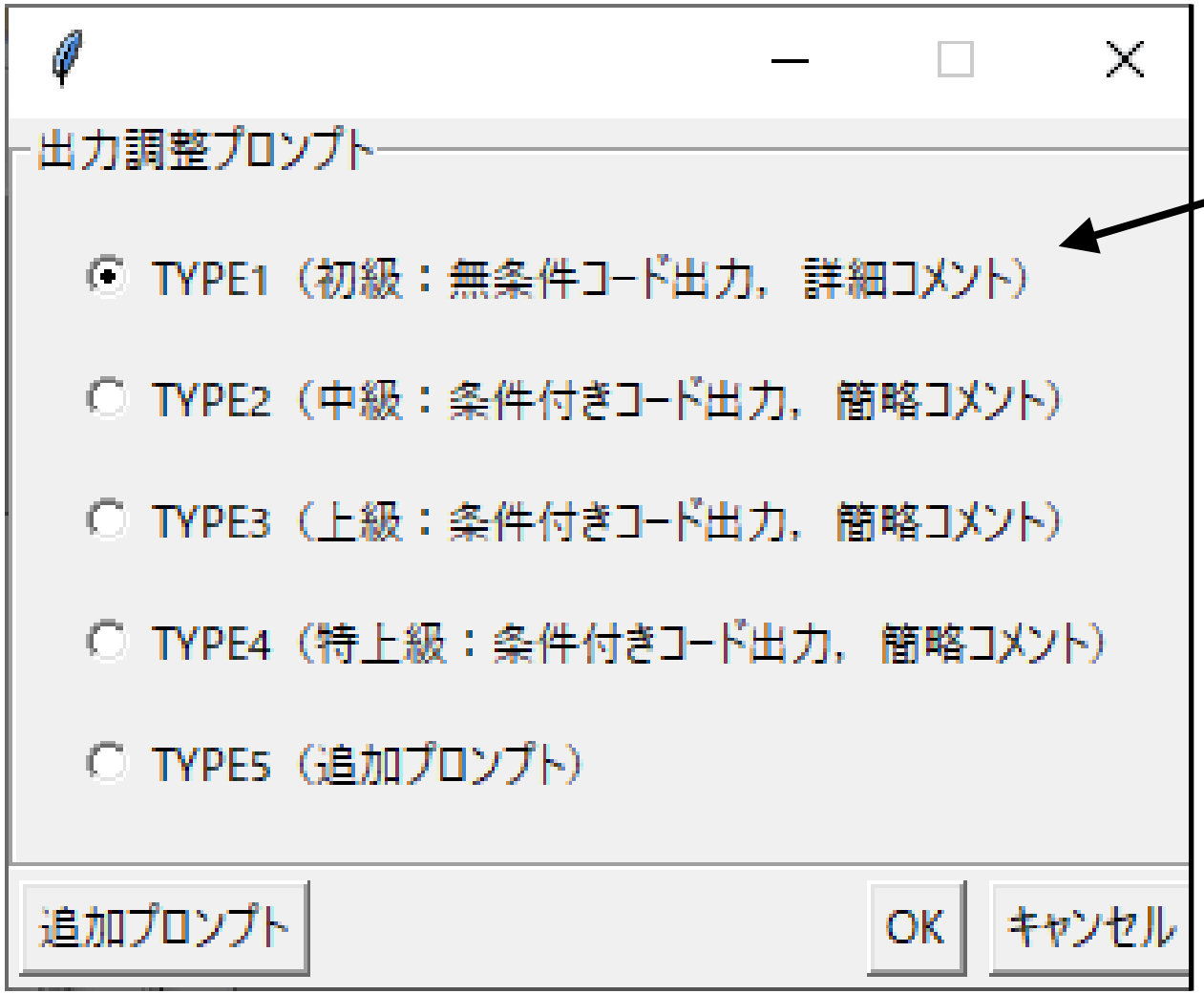
❼ マイコンボードへのコードの転送・実行

ハードウェアプロンプト



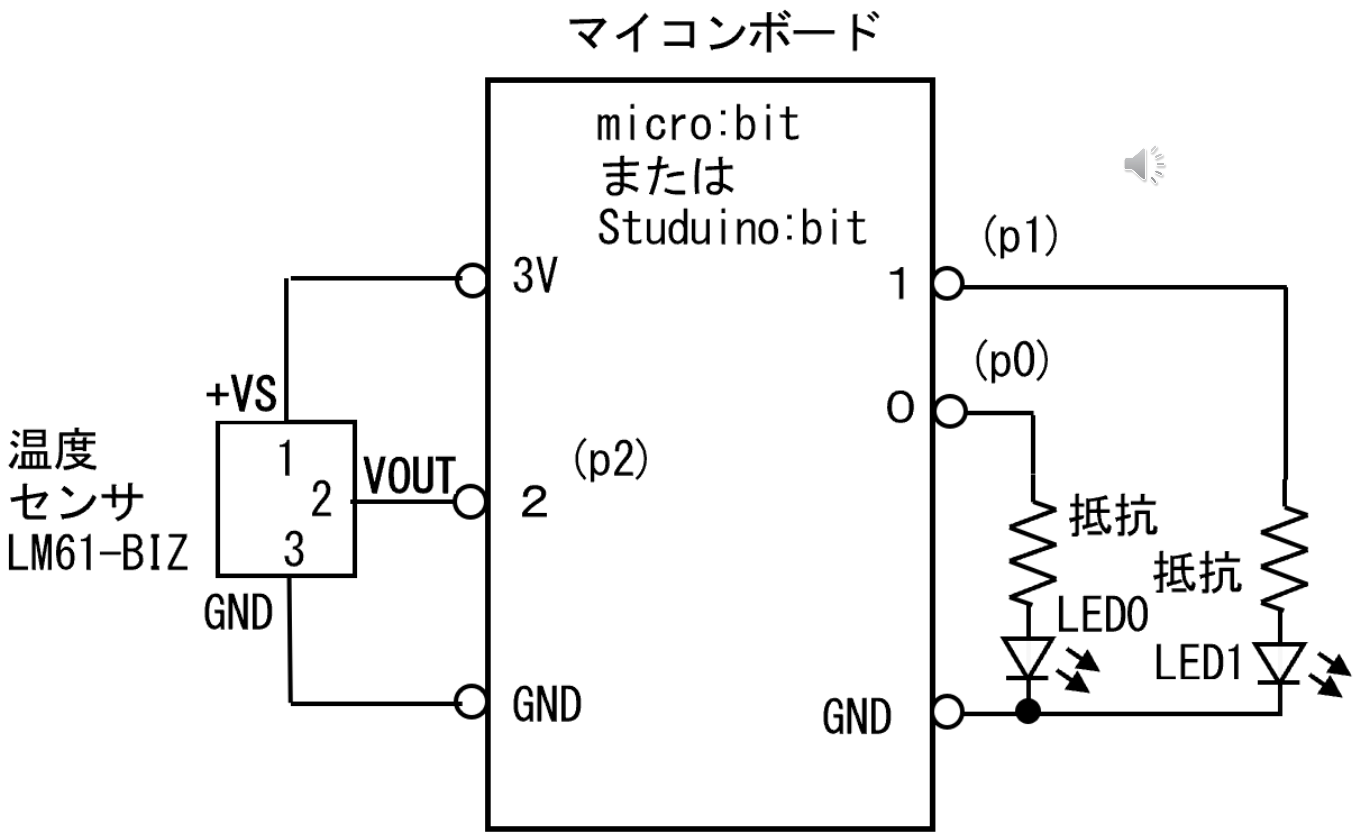
チェックボックスを選択することで，自動的にハードウェアプロンプトが入力される

出力調整プロンプト

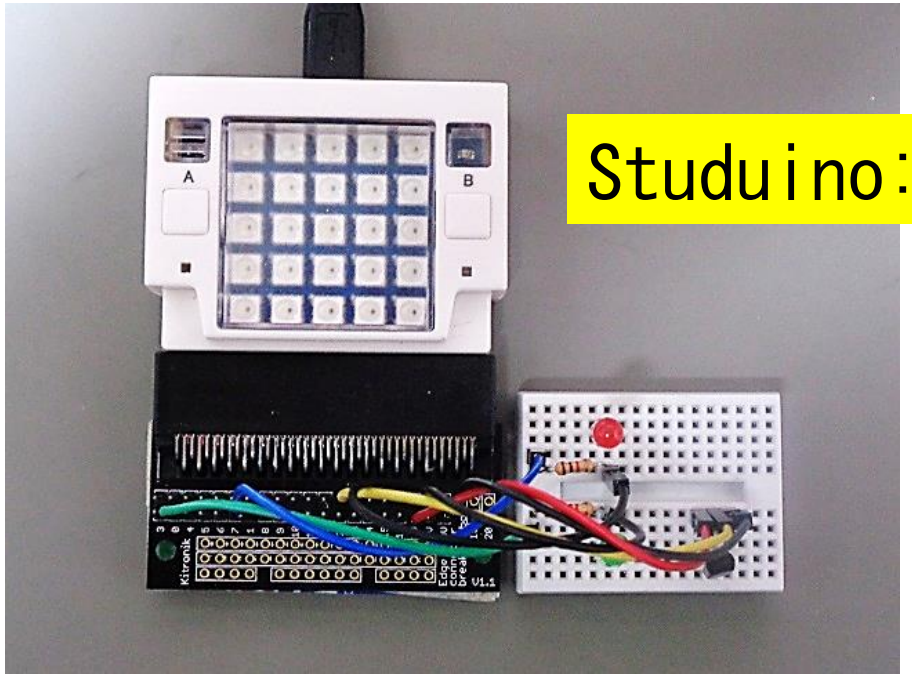


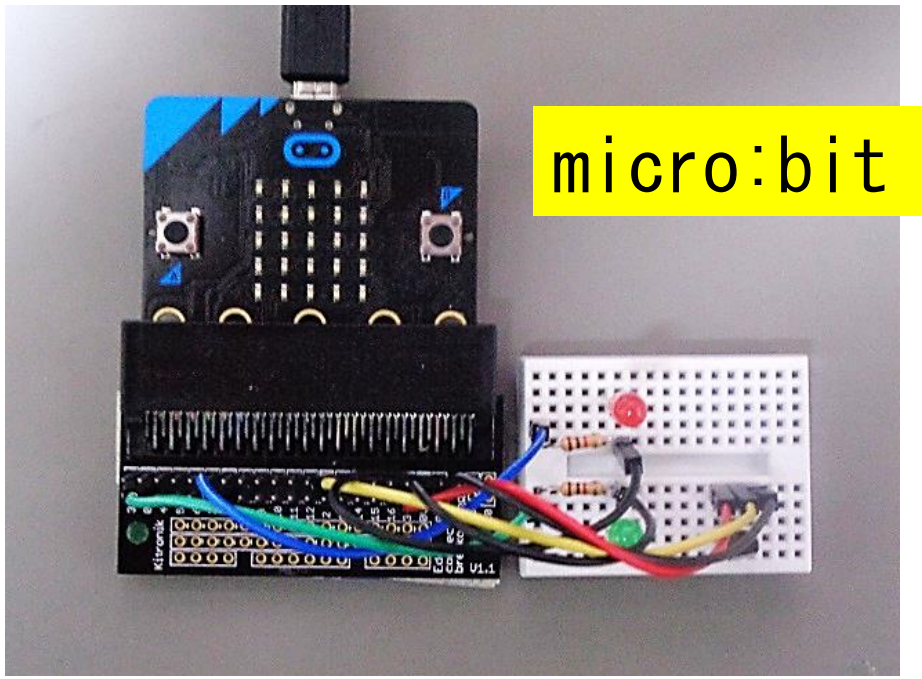
ラジオボタンを選択することで，自動的に出力調整プロンプトが入力される

実験例（温度の計測・制御）
テキスト入力



出力調整プロンプトはTYPE1





実験例（温度の計測・制御）ハードウェアプロンプト

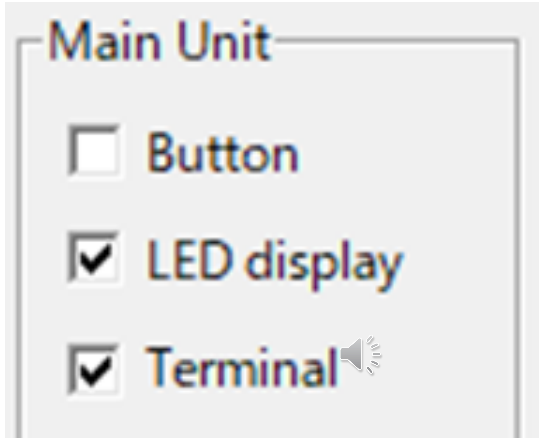
Pythonのバージョンは3.4です.
待ち時間はtime.sleep(秒単位)を使用してください.
"from sbblocks import *"で全ての機能を制御できます.

(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.

「LED display」と「Terminal」を選択するだけで, 自動的に入力される



実験例（温度の計測・制御）ハードウェアプロンプト

(2) Terminal（拡張端子）のハードウェアプロンプト

"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.

実験例（温度の計測・制御）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.

実験例（温度の計測・制御）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)
```

実験例（温度の計測・制御）生成されたコード

```
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>
```

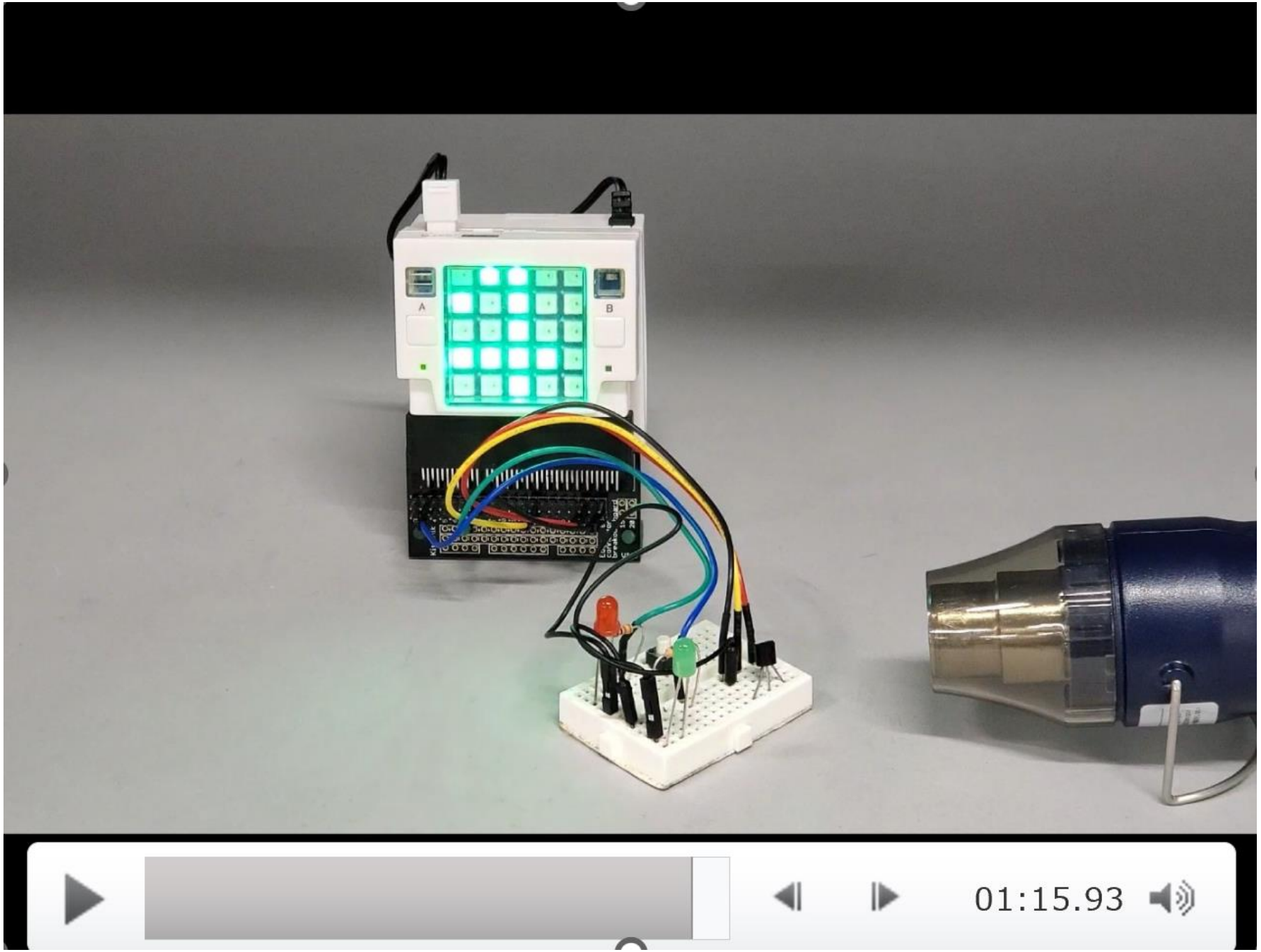
実験例（温度の計測・制御）生成されたコメント

<hint>
もしこのコードがあなたの意図したとおりの動作をしない場合，以下の点を確認してください：
1. 温度センサが正しくp2端子に接続されているか.
2. 温度センサの値が正しく読み取られているか.
3. 赤色LEDがp1端子に，緑色LEDがp0端子に接続されているか.
4. LEDが点灯するために必要な電源が供給されているか.
5. 温度の計算式が正しいか（センサの特性に基づいているか）.
6. 温度が30℃以上のときに赤色LEDが点滅すること，30℃未満のときに緑色LEDが点滅することを確認するために，実際の温度を測定してみてください.
</hint>

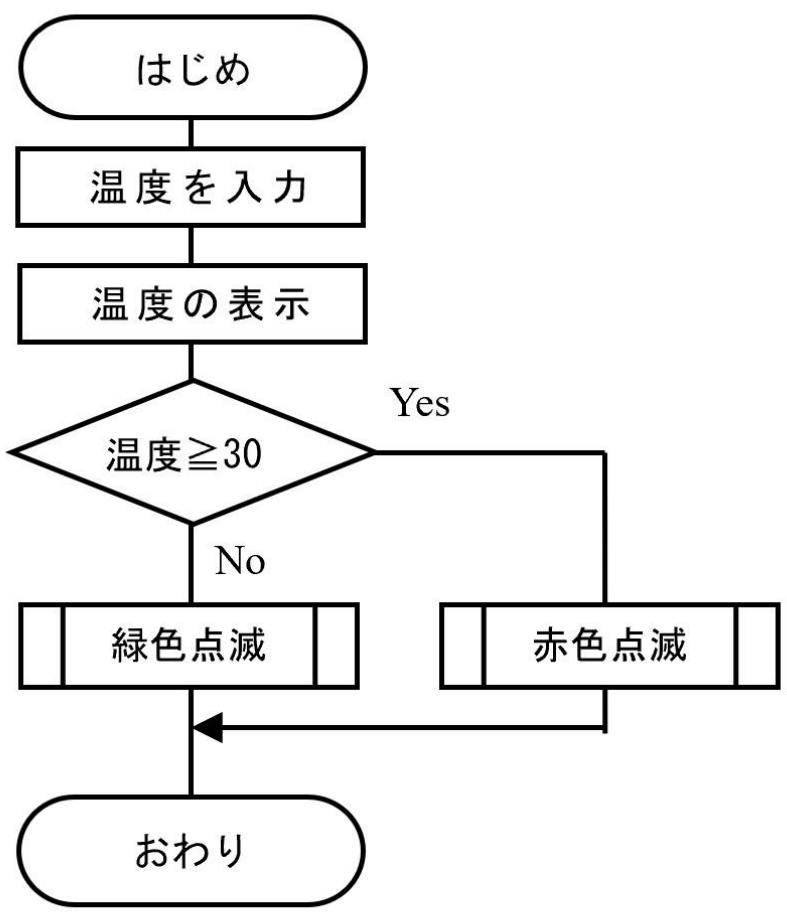
実験例

温度の
計測・制御
テキスト入力

動画



実験例（温度の計測・制御）フローチャート画像入力



実験例（温度の計測・制御）生成されたコード

```
<code>
from sbblocks import *
import time

def get_temperature(): # 温度を取得する関数
    sensor_value = p2.read_analog()
    temperature = 330 * sensor_value / 1023 - 60
    return temperature

def blink_green(): # 緑色LEDを点滅させる関数
    for _ in range(3):
        p0.write_digital(1)
        time.sleep(1)
        p0.write_digital(0)
        time.sleep(1)
```

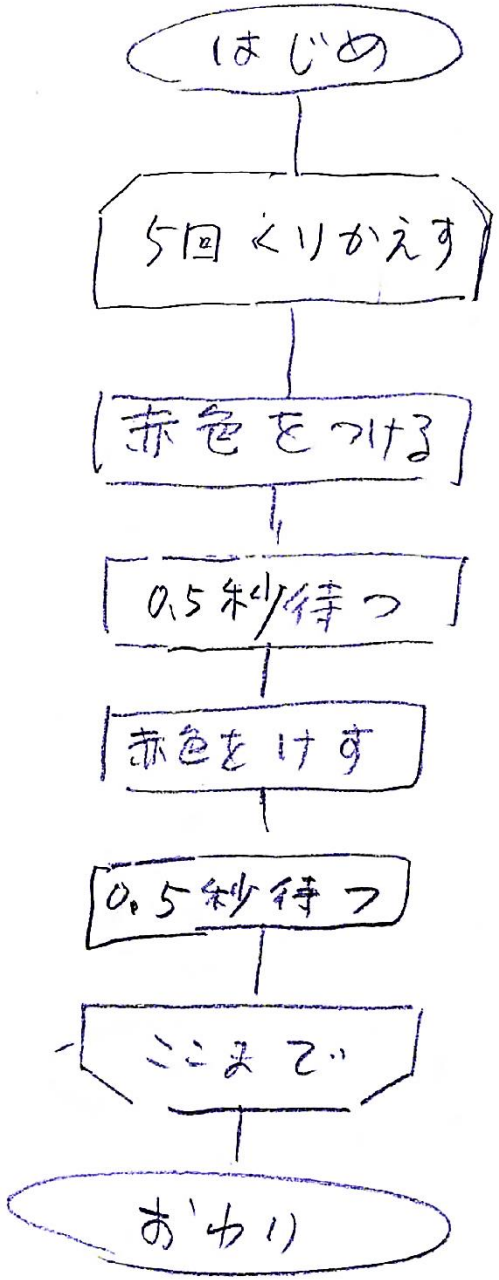
実験例（温度の計測・制御）生成されたコード

```
def blink_red(): # 赤色LEDを点滅させる関数
    for _ in range(5):
        p1.write_digital(1)
        time.sleep(0.5)
        p1.write_digital(0)
        time.sleep(0.5)

def main(): # メイン処理
    temperature = get_temperature()
    display.scroll("Temp: {:.1f}".format(temperature), delay=150)
    if temperature >= 30:
        blink_red()
    else:
        blink_green()
main()
</code>
```

実験例
（温度の計測・制御）

鉛筆の手書きによる
フローチャート画像
でも可能

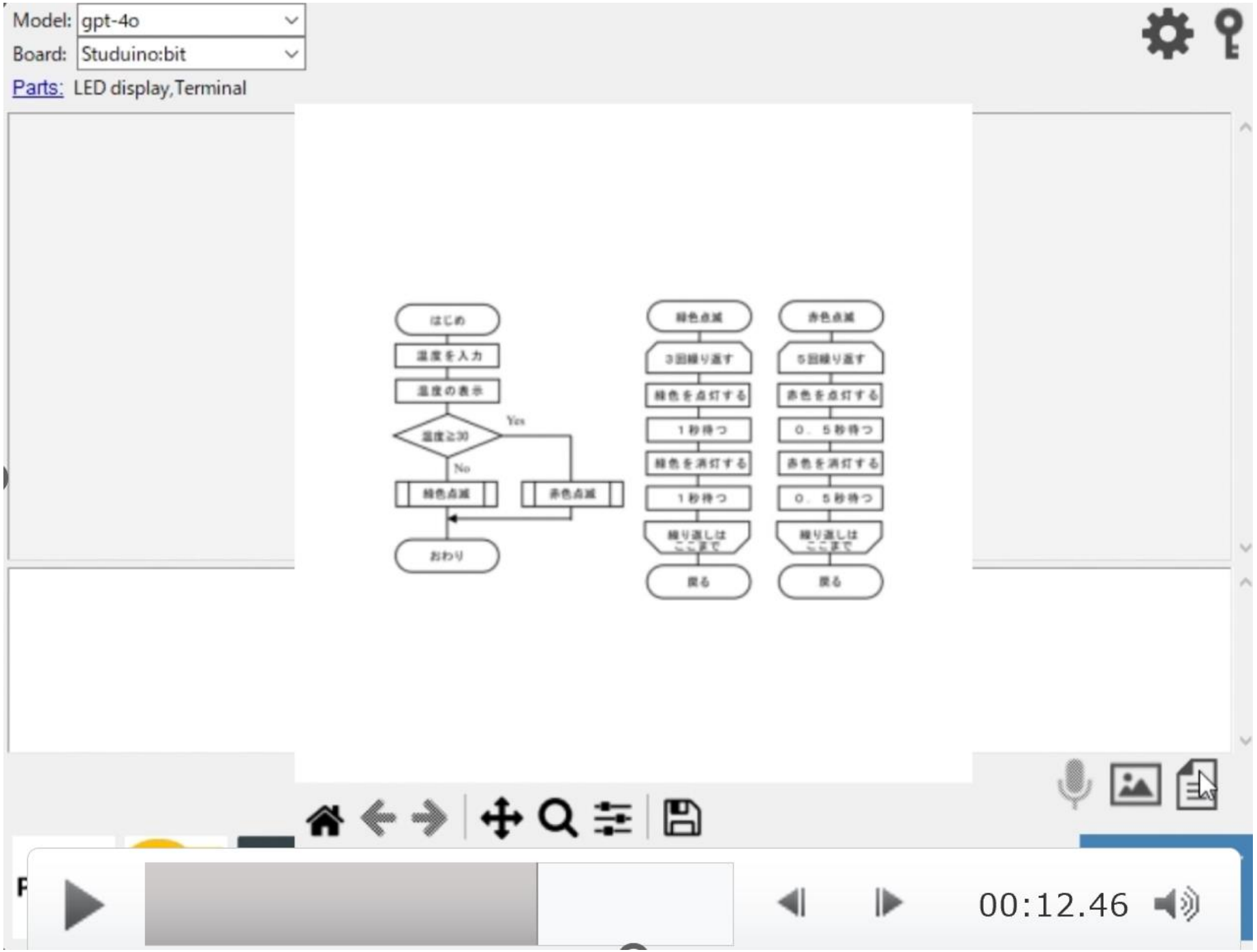
```
from sbblocks import *
import time

for i in range(5):
    pin1.write_digital(1)
    time.sleep(0.5)
    pin1.write_digital(0)
    time.sleep(0.5)
```



実験例

温度の
計測・制御
フローチャート
画像入力

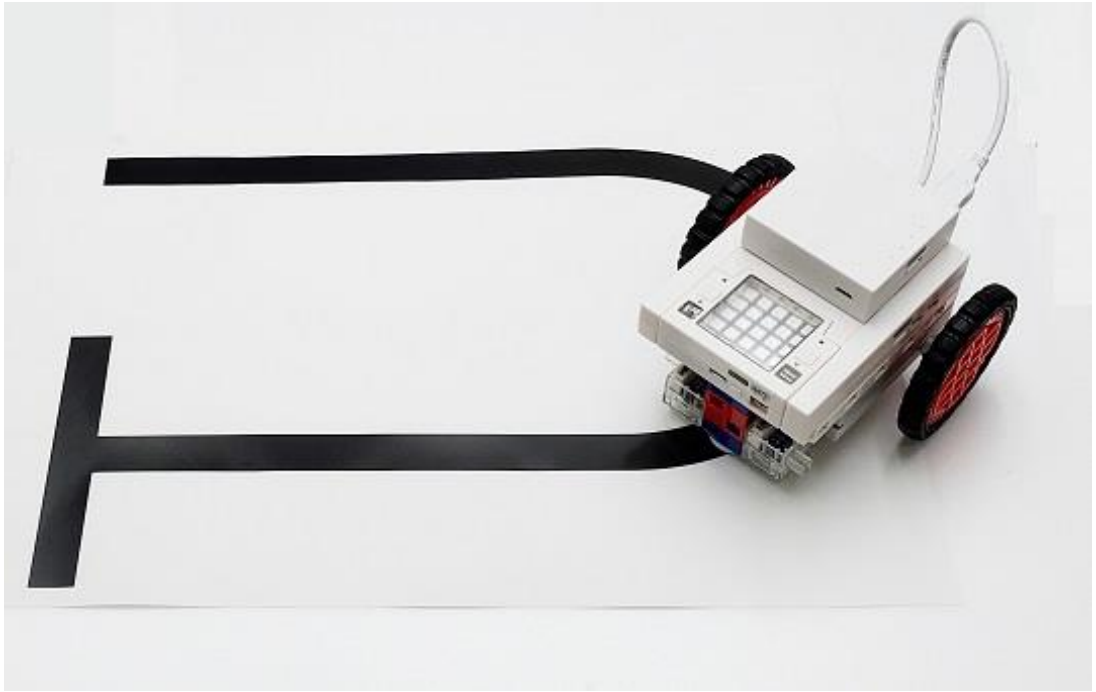
動画



実験例（ライントレースロボット）テキスト入力

ライントレースロボットとは，黒線に沿って自動走行する車型ロボット

本実験では，白い紙に描かれた黒線をたどるため，2個の赤外線フォトリフレクタと2個のDCモータを使用



出力調整プロンプトはTYPE1

実験例（ライントレースロボット）ハードウェアプロンプト

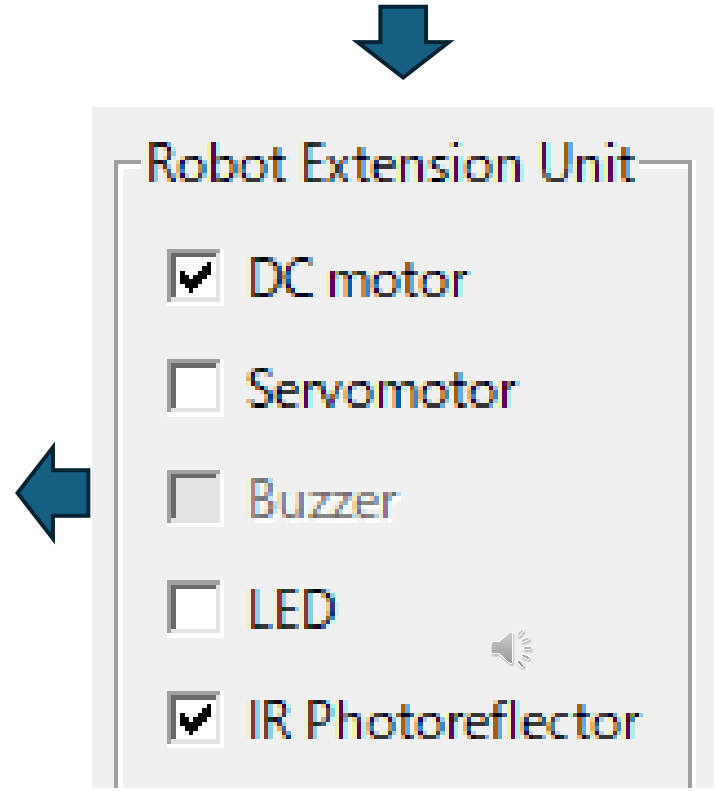
Pythonのバージョンは3.4です.
待ち時間はtime.sleep(秒単位)を使用してください.
"from sbblocks import *"で全ての機能を制御できます.

(1)DC motorのハードウェアプロンプト

"DCモータ"は, DCMotorクラスで制御できます. DCMotorクラスは次のメソッドだけです.

- __init__(pin): pin引数に M1かM2を文字列で指定してDCモータのインスタンスを生成します.
- power(power): power引数にパワー(0～100)を指定してDCモータのパワーを設定します.
- cw(): DCモータを正転します.
- ccw(): DCモータを逆転します.
- stop(): DCモータを解放します.
- brake(): DCモータを停止します.

「DC motor」と「IR PhotoRefrector」を選択するだけで, 自動的に入力される



実験例（ライントレースロボット）ハードウェアプロンプト

(2)赤外線フォトリフレクタのハードウェアプロンプト

"赤外線フォトリフレクタ"は，IRPhotoReflectorクラスで制御できます．IRPhotoReflectorクラスは次のメソッドだけです．

・__init__(pin)：pin引数に P0かP1かP2を文字列で指定して赤外線フォトリフレクタのインスタンスを生成します．

・get_value()：センサの値(0～100)を取得します．

実験例（ライントレースロボット）ユーザ作成プロンプト

[部品の説明]

DCモータはM1,M2に接続しています.

赤外線フォトリフレクタはP0,P1に接続しています.

[動きの説明]

次の動きを実現してください.

P0の赤外線フォトリフレクタが50よりも小さい場合，M1のDCモータをパワー50で逆転しM2のDCモータを止める

P1の赤外線フォトリフレクタが50よりも小さい場合，M1のDCモータを止めてM2のDCモータをパワー50で逆転する

それ以外は，M1とM2のDCモータをパワー50で逆転する

実験例（ライントレースロボット）生成されたコード

```
<code>
from sbblocks import *
import time
# DCモータのインスタンスを生成
motor1 = DCMotor("M1")
motor2 = DCMotor("M2")
# 赤外線フォトリフレクタのインスタンスを生成
sensor0 = IRPhotoReflector("P0")
sensor1 = IRPhotoReflector("P1")
# モータのパワーを設定
motor_power = 50
while True:
    # センサの値を取得
    value0 = sensor0.get_value()
    value1 = sensor1.get_value()
```

実験例（ライントレースロボット）生成されたコード

```
if value0 < 50:
        motor1.power(motor_power)
        motor1.ccw()  # M1を逆転
        motor2.stop()  # M2を停止
    elif value1 < 50:
        motor1.stop()  # M1を停止
        motor2.power(motor_power)
        motor2.ccw()  # M2を逆転
    else:
        motor1.power(motor_power)
        motor1.ccw()  # M1を逆転
        motor2.power(motor_power)
        motor2.ccw()  # M2を逆転
time.sleep(0.1)  # 100ms待機
</code>
```

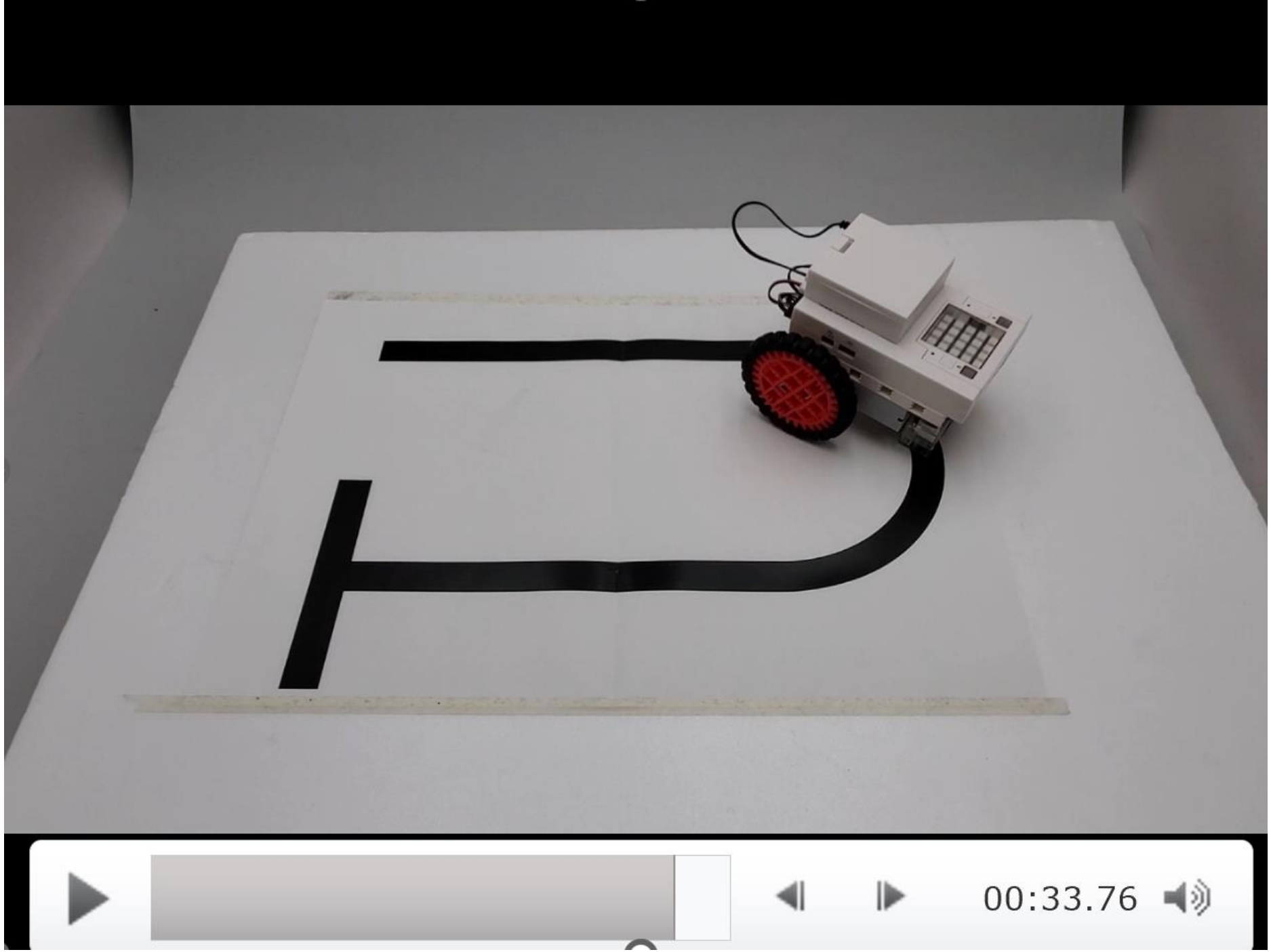
実験例（ライントレースロボット）生成されたコメント

<hint>
もしこのプログラムがあなたの意図したとおりの動作をしない場合，以下の点を確認してください.
1. DCモータの出力値は50に設定されていますが，モータの動作が弱い場合はこの値を調整してください.
2. M1とM2のDCモータはそれぞれ正しいピンに接続されていることを確認してください.
3. 赤外線フォトリフレクタの閾値は50に設定されていますが，環境によってはこの値を変更する必要があるかもしれません.
4. 赤外線フォトリフレクタは正しく取り付けられ，適切に動作していることを確認してください.
5. モータの回転方向は，逆転がロボットの後ろに進むことを意味しますので，動作が期待通りでない場合はモータの接続を見直してください.
</hint>

実験例

ライントレース
ロボット

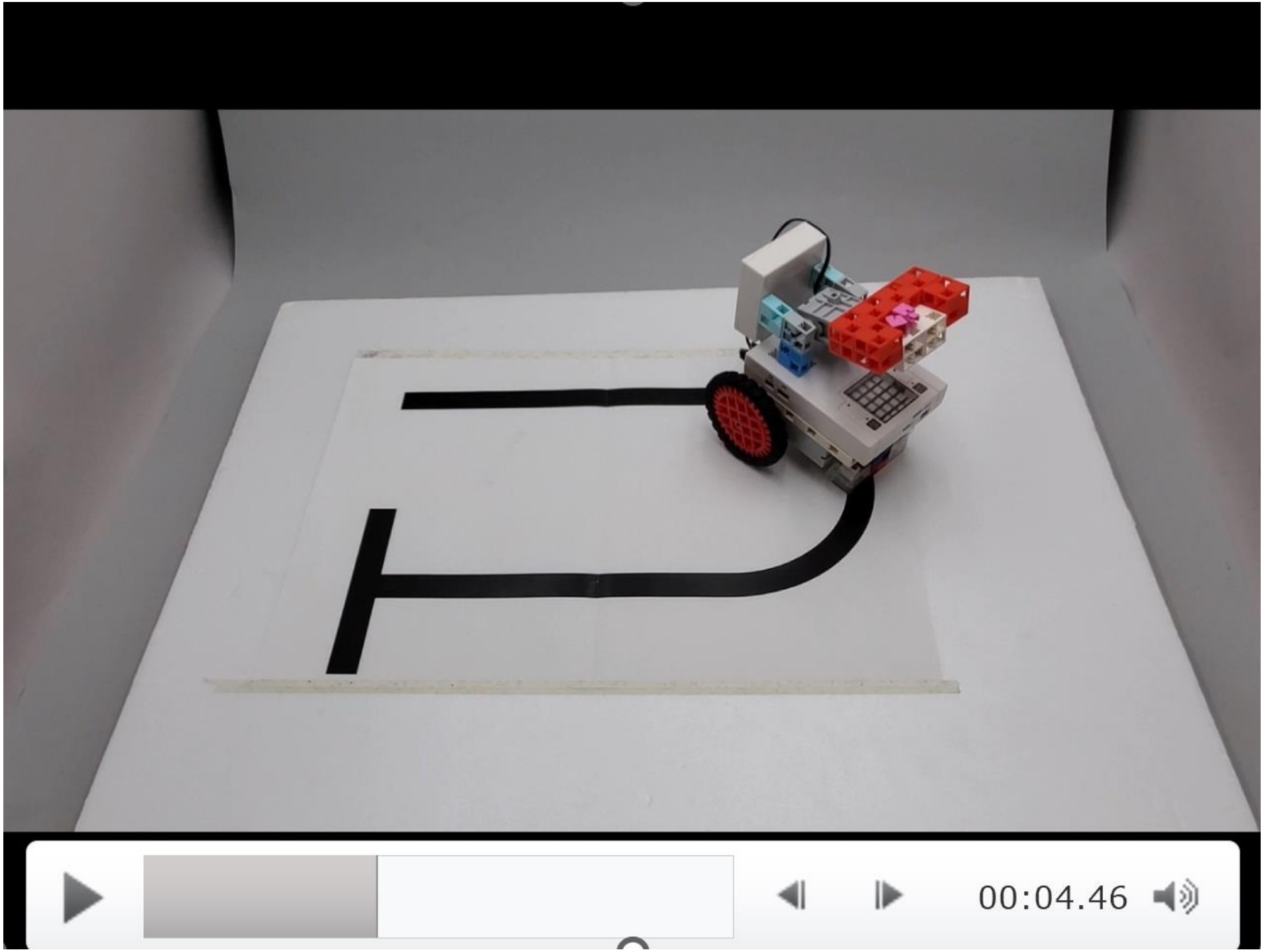
動画



実験例

搬送
ロボット

動画

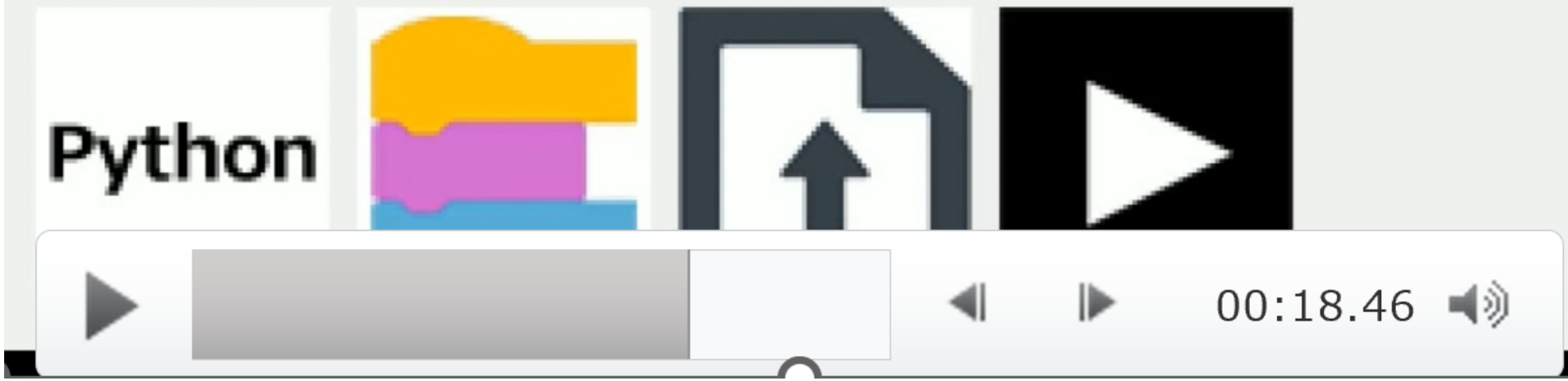


実験例

LED回路
音声入力

赤色のLEDを0.5秒間隔で5回点滅してください

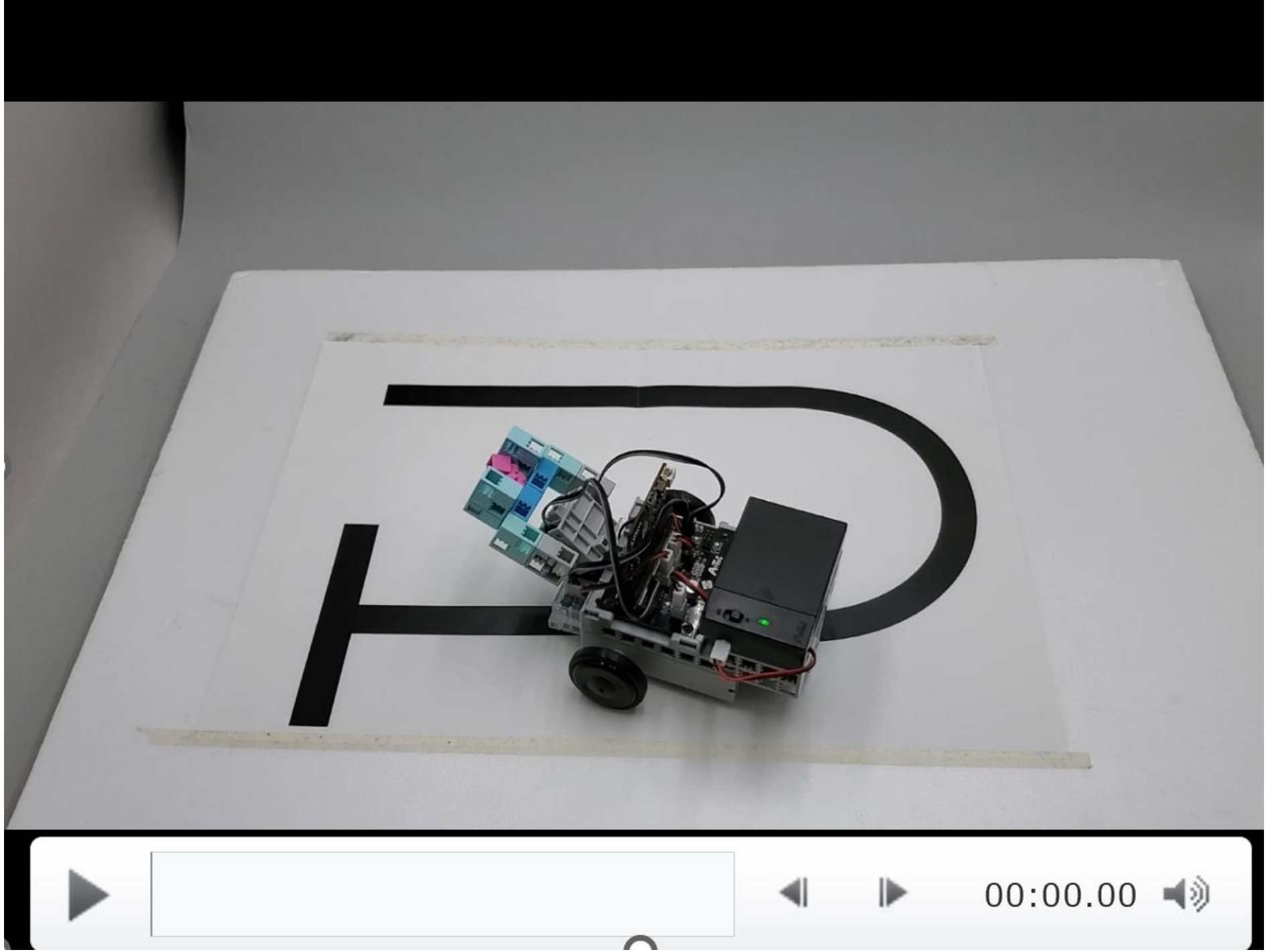
動画



実験例

マイクロビットによる
搬送ロボット

動画



実験例

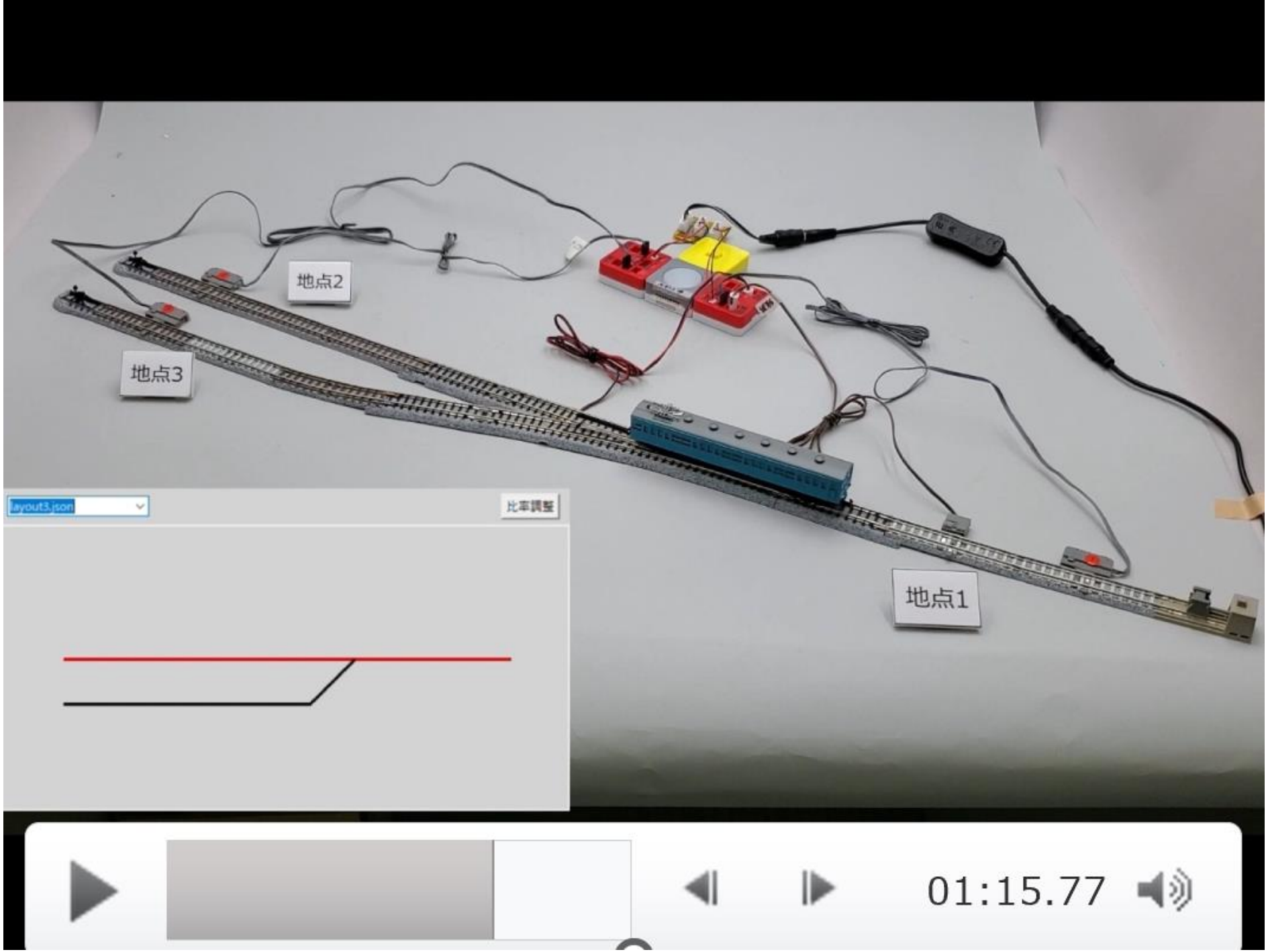
LEDディスプレイ

動画

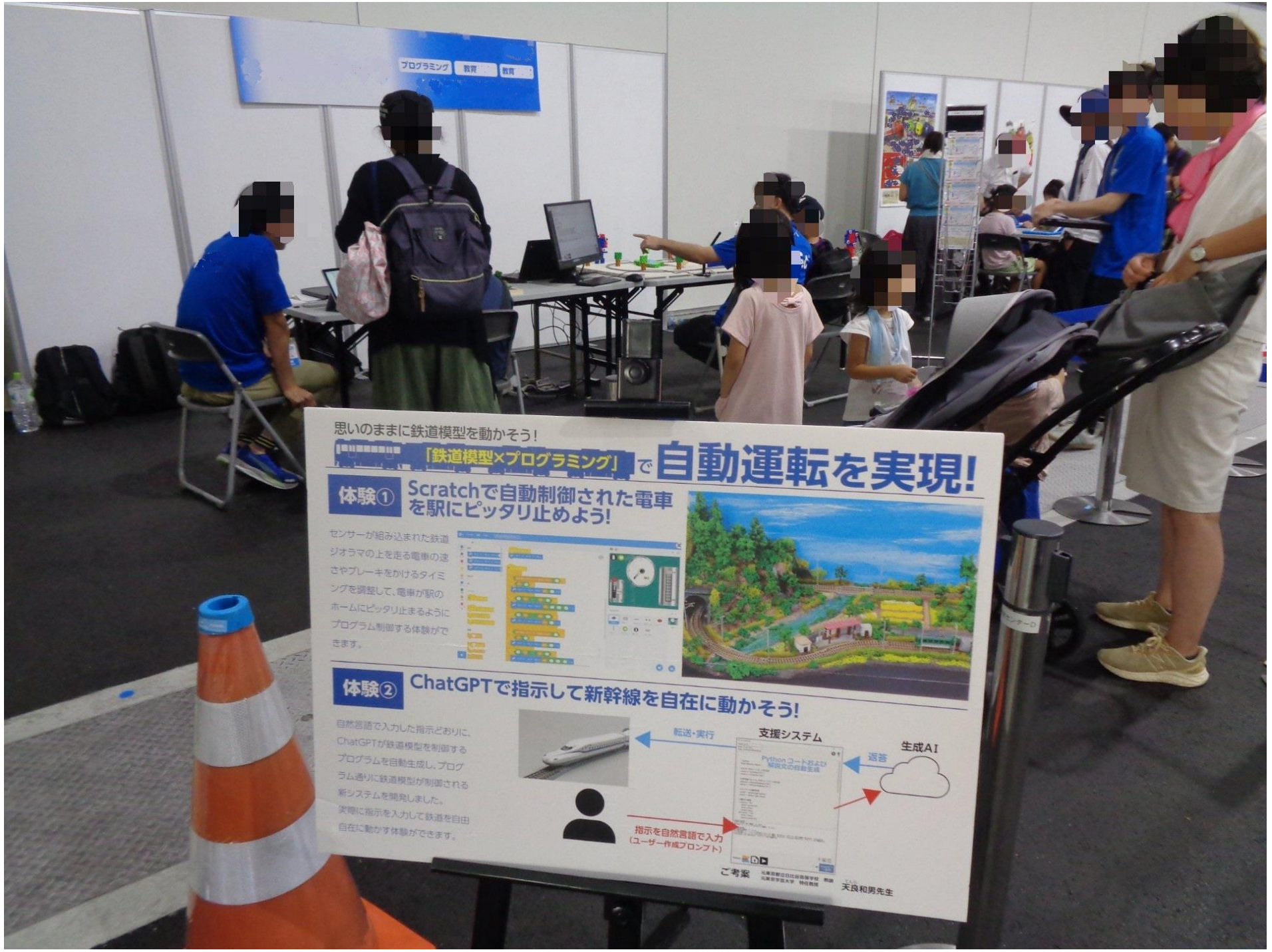
実験例

ArtecLinksによる鉄道模型制御

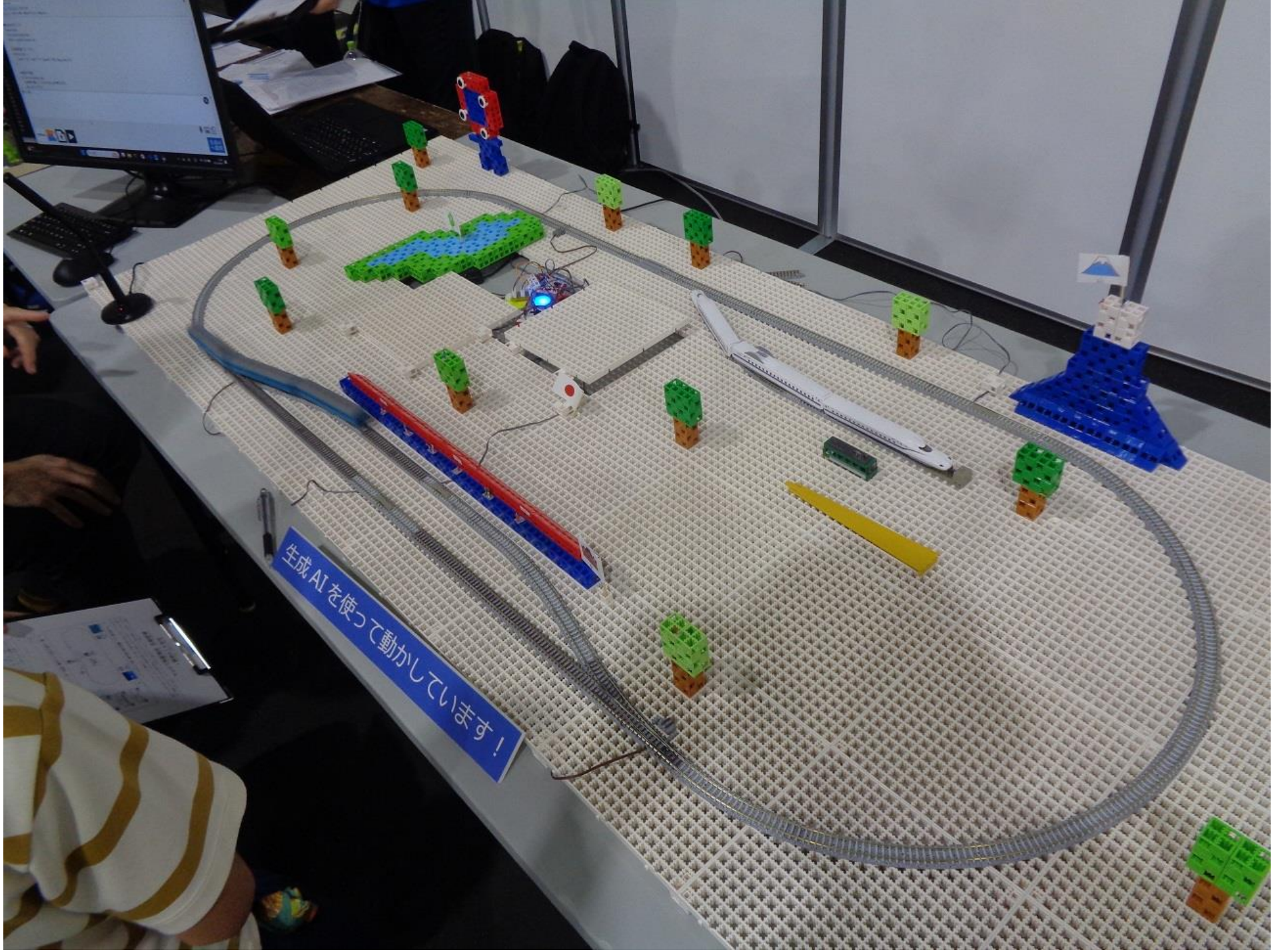
動画



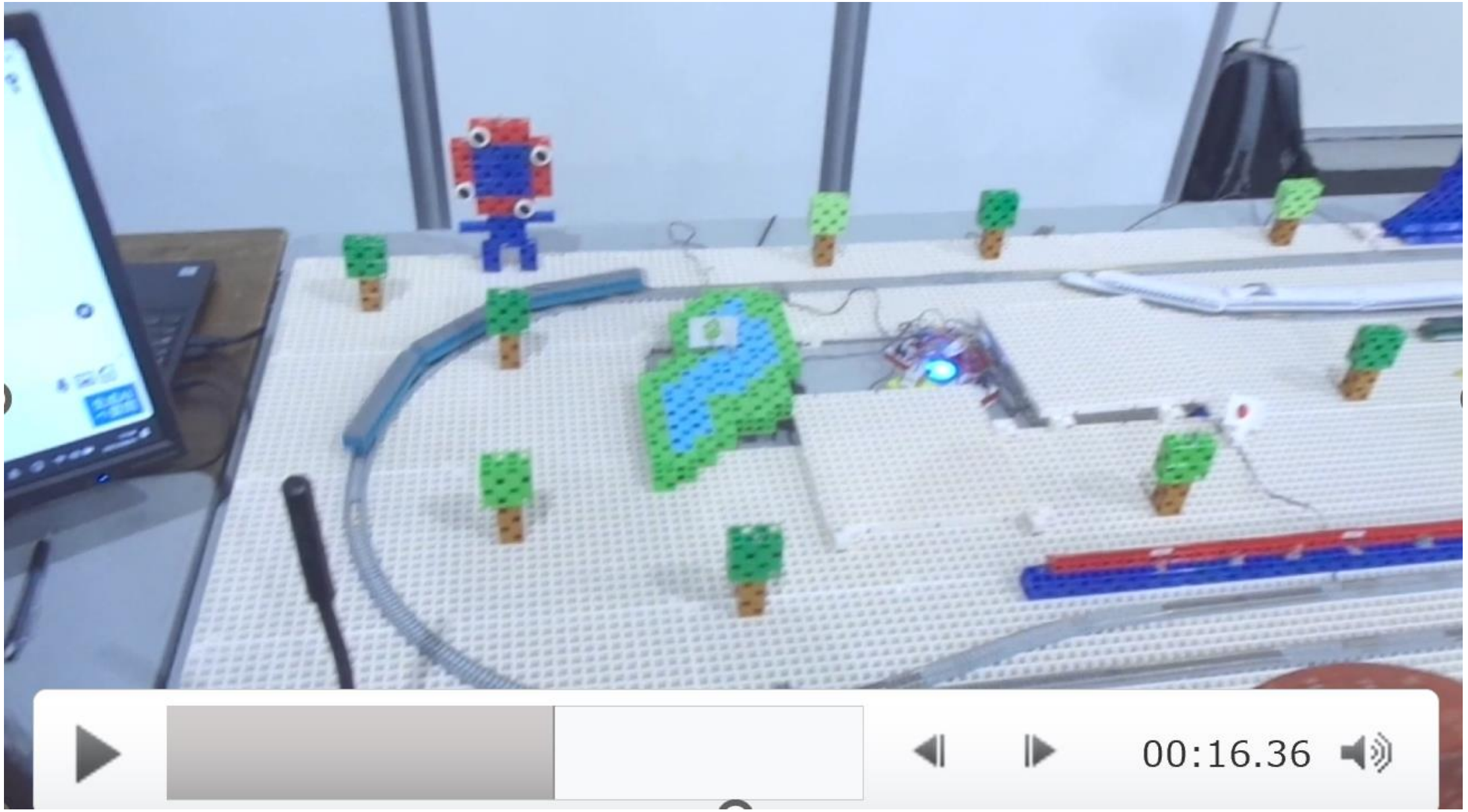
万博展示


思いのままに鉄道模型を動かそう!
「鉄道模型×プログラミング」で自動運転を実現!
体験①
Scratchで自動制御された電車を駅にピッタリ止めよう!
センサーが組み込まれた鉄道ジオラマの上を走る電車の速さやブレーキをかけるタイミングを調整して、電車が駅のホームにピッタリ止まるようにプログラム制御する体験ができます。
体験②
ChatGPTで指示して新幹線を自在に動かそう!
自然言語で入力した指示どおりに、ChatGPTが鉄道模型を制御するプログラムを自動生成し、プログラム通りに鉄道模型が制御される新システムを開発しました。
実際に指示を入力して鉄道を自由自在に動かす体験ができます。
転送・実行
支援システム
生成AI
返答
指示を自然言語で入力
(ユーザー作成プロンプト)
ご考案
天良和男先生

万博展示



万博展示　動画



ARCSモデルによるアンケート

- 大阪・関西万博で実施された生成AIを用いた鉄道模型プログラミング体験イベントにおいて，体験後にARCSモデルに基づくアンケートを実施
- ARCSモデルとは
  - ケラーが提唱した学習者の動機づけを高めるためのモデル
  - 教材開発や教育評価にも活用されている
- ARCSの4要素と下位項目
  - 学習者のやる気を引き出す4つの要素

    注意（Attention），関連性（Relevance）
    自信（Confidence），満足感（Satisfaction）
  - 各要素には下位項目がある

    A1～A3，R1～R3
    C1～C3，S1～S3

ARCSモデルの下位項目

参考文献：「教育工学事典」
日本教育工学会，実教出版，2000

項目	下位項目
A 注意	A1（知覚的喚起）学習者の興味をひくため何ができるか
	A2（探究心の喚起）探求の態度を刺激できるか
	A3（変化性）学習者の注意を維持できるか
R 関連性	R1（親しみやすさ）学習者の経験と教材とを結びつけることができるか
	R2（目的指向性）学習者の目的と教材を関連づけられるか
	R3（動機との一致）学習者の学習様式や興味と教材とを関連づけられるか
C 自信	C1（学習欲求）学習者が前向きな成功への期待感を持つように支援できるか
	C2（成功の機会）学習経験が，学習者自身の能力に対する信念を支えたり高めたりするのか
	C3（コントロールの個人化）学習者は自分の成功が自分の努力と脳力によるものであると確信するか
S 満足感	S1（内発的強化）学習経験の本来の楽しみを促進し支援できるか
	S2（外発的報酬）学習者の成功に対して，報酬的結末を提供するのか
	S3（公平さ）学習者が公平に扱われていると感じるか

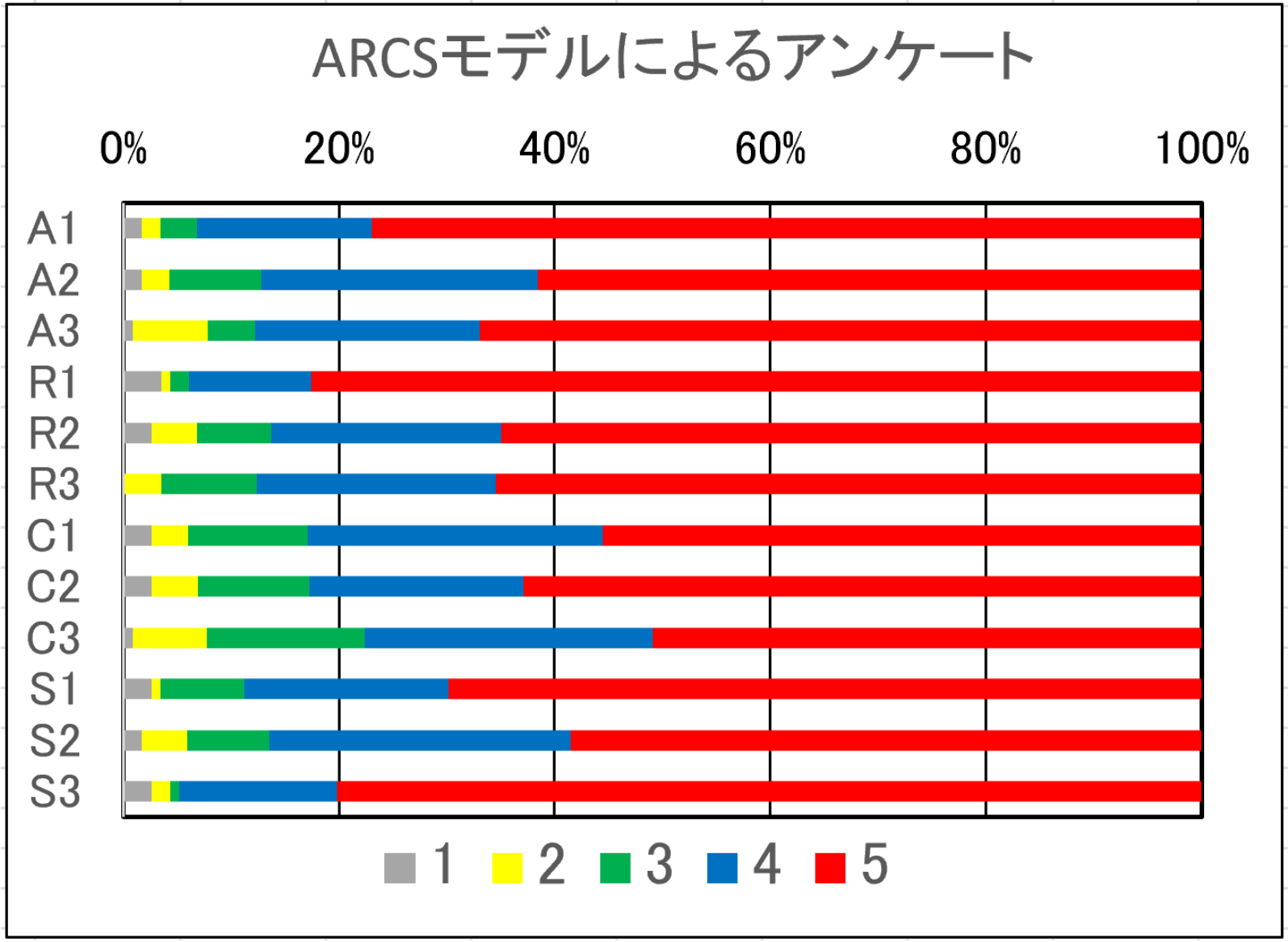
ARCSモデルによるアンケート

番号	質問
A1	電車が自動で走ったり、ポイント（わかれ道）が切りかわったりするようすを見て、おもしろいと感じましたか。
A2	生成AI（せいせいエーアイ）に伝える文（プロンプト）を入力して、電車を動かせるしくみをもっと知りたいと思いましたか。
A3	電車の動きやポイント（分かれ道）の変化に注目して、さいごまで集中して取りくめましたか。
R1	むずかしいプログラミングのことばを知らなくても、日本語だけで電車を動かせたので、安心して楽しく体験できましたか。
R2	この体験は、これからの勉強や仕事に役立つと思いましたか。
R3	この体験は、あなたの好きなこと（たとえば鉄道・ロボット・プログラミング）に合っていると感じましたか。

ARCSモデルによるアンケート

番号	質問
C1	電車を動かすために、生成AI（せいせいエーアイ）に伝える文（プロンプト）をどのように書けばいいかを考えたり、いろいろ試してみたりして、もっとがんばってみたいと思いましたか。
C2	生成AI（せいせいエーアイ）に伝える文（プロンプト）で電車がうまく動いたとき、「やればできる」と思いましたか。
C3	電車がうまく動いたとき、「自分で工夫したからできたんだ」と思って、自信がつきましたか。
S1	この体験が楽しくて、「もっとやってみたい」と思いましたか。
S2	電車が思ったとおりに動いたとき、生成AI（せいせいエーアイ）が「その文は正しい」と認めてくれたように感じましたか。
S3	プログラミングや鉄道模型について知らなくても、みんなが楽しめるように工夫されていたと感じましたか。

ARCSモデルによるアンケート



1 全くそう思わない
2 あまりそう思わない
3 どちらとも言えない
4 まあそう思う
5 とてもそう思う

n = 116

- 各項目において高評価が得られている
- この活動は，学習者の関心を喚起し，学習意欲の向上に寄与している

出力調整プロンプトの違いによる出力変化

❶　❺
コードが生成されるが，間違ったコードが生成される

黄色の部分はコード生成あり		出力調整プロンプト			
		TYPE1	TYPE2	TYPE3	TYPE4
プログラム作成	USER1	❶	②	③	④
	USER2	❺	⑥	⑦	⑧
	USER3	⑨	⑩	⑪	⑫
	USER4	⑬	⑭	⑮	⑯

	部品の説明	動きの説明
USER1	なし	LEDを点灯してください
USER2	LED	LEDを点灯してください
USER3	LEDをp1に接続する	LEDを点灯してください
USER4	LEDをp1に接続する	「p1.write_digital(1)」を使ってLEDの点灯してください

❻ 具体的で詳細なアルゴリズムを指示したものだけがコードが生成される

PythonからScratchへの変換機能

生成AIが出力したPythonコードを構文解析し，Scratchに変換

```
# DCモータのインスタンスを生成
motor1 = DCMotor("M1")
motor2 = DCMotor("M2")

# 赤外線フォトリフレクタのインスタンスを生成
sensor0 = IRPhotoReflector("P0")
sensor1 = IRPhotoReflector("P1")

# センサーの値を取得
value0 = sensor0.get_value()
value1 = sensor1.get_value()

# 動作の実現
if value0 < 50:
    motor1.power(50)
    motor1.ccw()
    motor2.stop()
elif value1 < 50:
    motor1.stop()
    motor2.power(50)
    motor2.ccw()
else:
    motor1.power(50)
    motor1.ccw()
    motor2.power(50)
    motor2.ccw()
```

「Pythonにはあるが，Scratchにはないコードを出力しないでください」といったプロンプトをユーザ作成プロンプトに追加する必要がある

プロンプトリテラシー（質問力）の育成

- 生成AI時代においては，自分の意図や目的を的確に伝え，生成AIから適切な出力を得るための質問力（プロンプトリテラシー）が重要である
- 本システムを活用することで，ロボットや鉄道模型などの実物の動きを通して，「プロンプトリテラシー」や「プログラミング的思考」を育むことができる

おわりに

- 他のマイコンボードなど多様な環境への適用
- 出力調整プロンプトの改良などプロンプト機能の強化
- マイコンボードなしの環境での出力調整プロンプトの構築
- 実際の授業での活用と本システムを利用した学習者の評価