

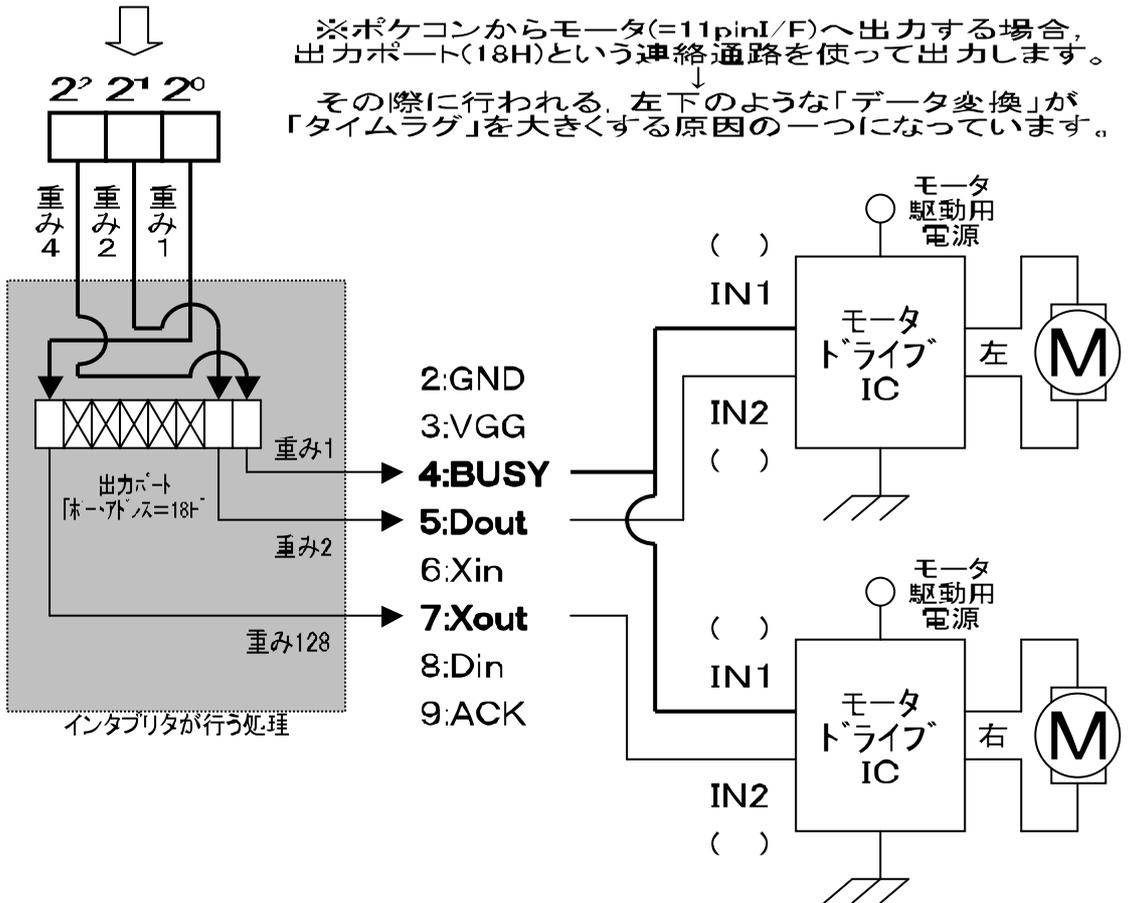
選択課題 2

マシン語による制御

～よりスムーズな制御のために～

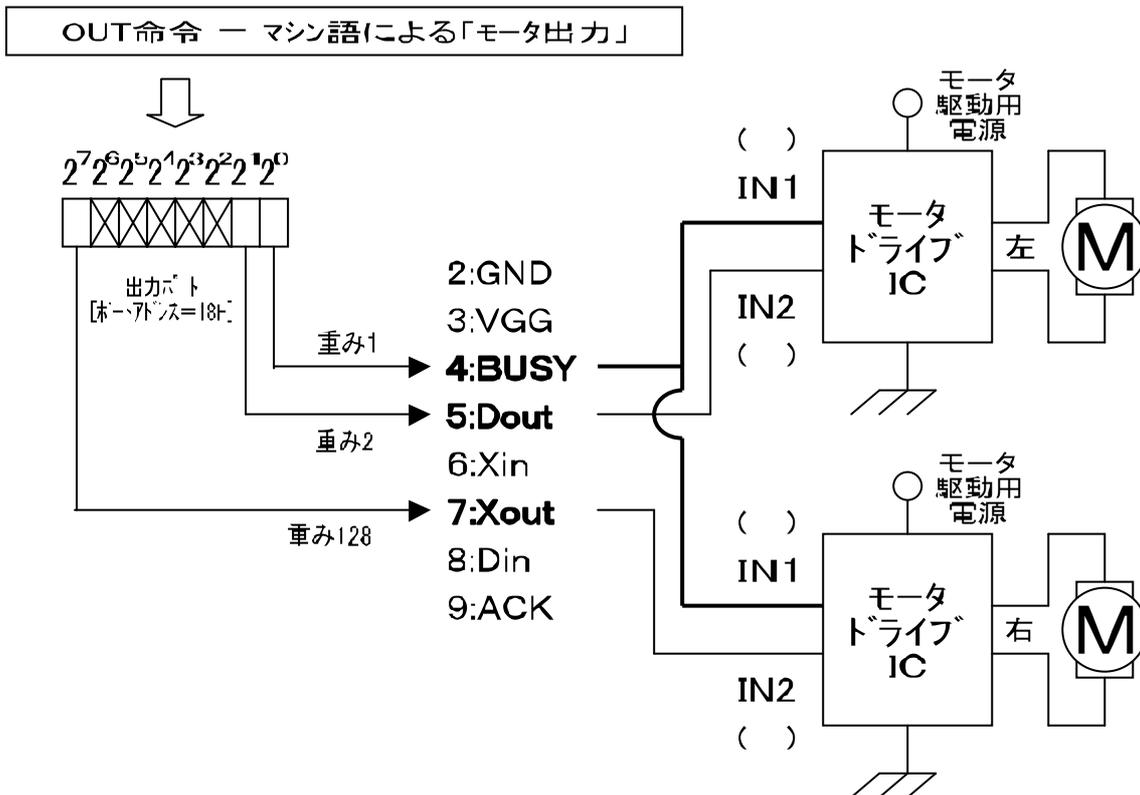
「モータ出力」の仕組み

OUT命令 = BASICによる「モータ出力」



OUT命令	出力データ	左モータの動作	右モータの動作
OUT 0	(0 0 0) ₂	フリー	フリー
OUT 1	(0 0 1) ₂	フリー	逆転
OUT 2	(0 1 0) ₂	逆転	フリー
OUT 3	(0 1 1) ₂	逆転	逆転
OUT 4	(1 0 0) ₂	正転	正転
OUT 5	(1 0 1) ₂	正転	ブレーキ
OUT 6	(1 1 0) ₂	ブレーキ	正転
OUT 7	(1 1 1) ₂	ブレーキ	ブレーキ

問題 左ページと次の図を参考に，下の表を完成させよう。出力データは，16進数で書くこと。 ヒント：pokeC を実行してみよう



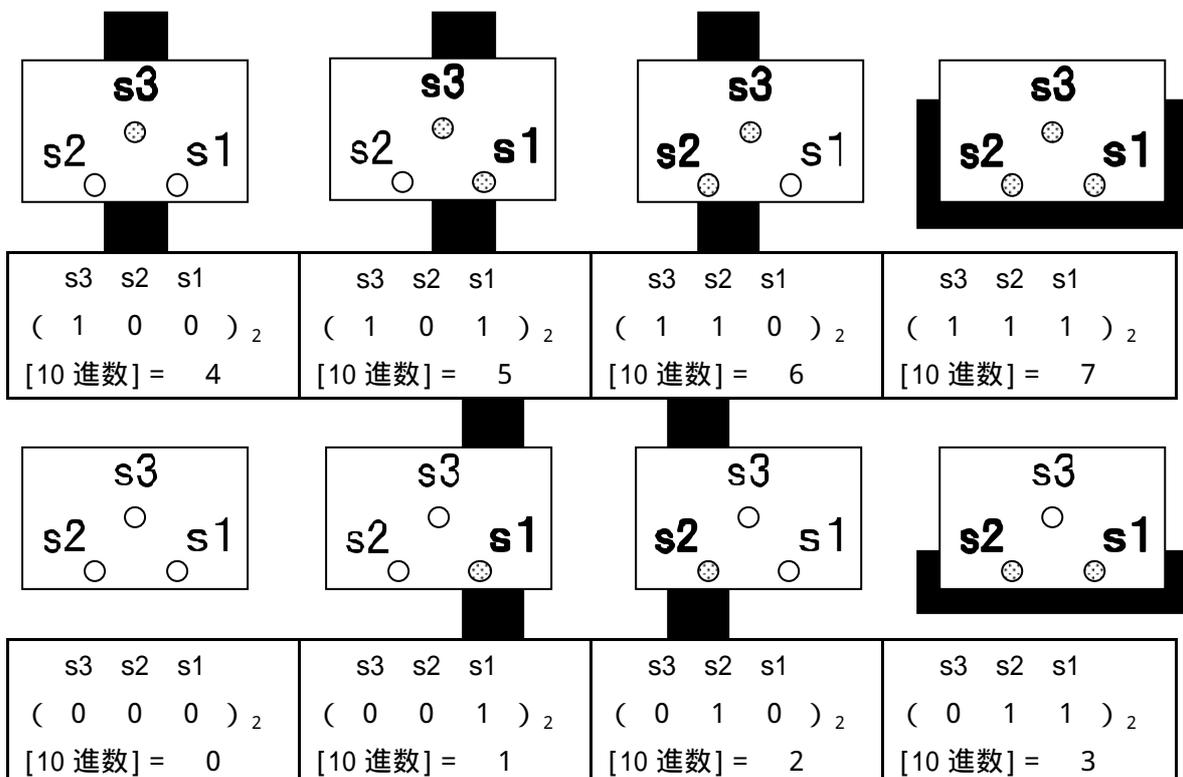
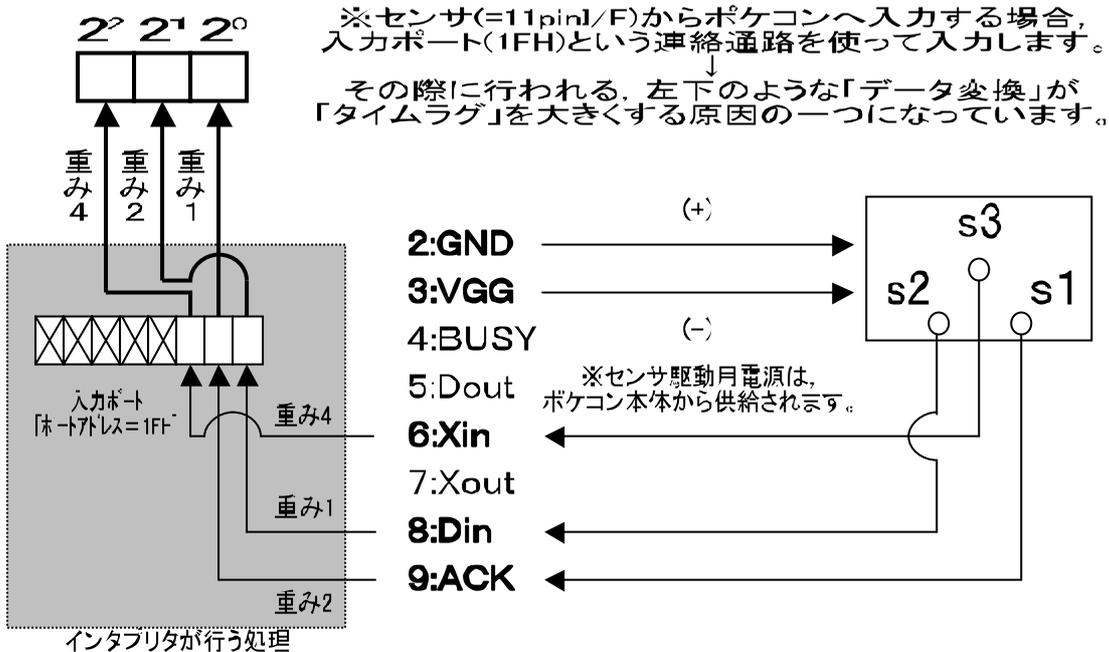
モータ駆動用電源とは，ポケモン搬送車の「乾電池」のことです。

マシン語の「OUT命令」は，出力ポート (= 11pin 端子) に直接出力するため，「タイムラグ」は限りなくゼロに近づきますが，出力データの形式がBASICと異なるので注意が必要です。 [4:BUSY = 重み 1, 5:Dout = 重み 2, 7:Xout = 重み 128]

BASIC と比較する	出力データ	左モータの動作	右モータの動作
OUT 0 に相当	() ₁₆	フリー	フリー
OUT 1 に相当	() ₁₆	フリー	逆転
OUT 2 に相当	() ₁₆	逆転	フリー
OUT 3 に相当	() ₁₆	逆転	逆転
OUT 4 に相当	() ₁₆	正転	正転
OUT 5 に相当	() ₁₆	正転	ブレーキ
OUT 6 に相当	() ₁₆	ブレーキ	正転
OUT 7 に相当	() ₁₆	ブレーキ	ブレーキ

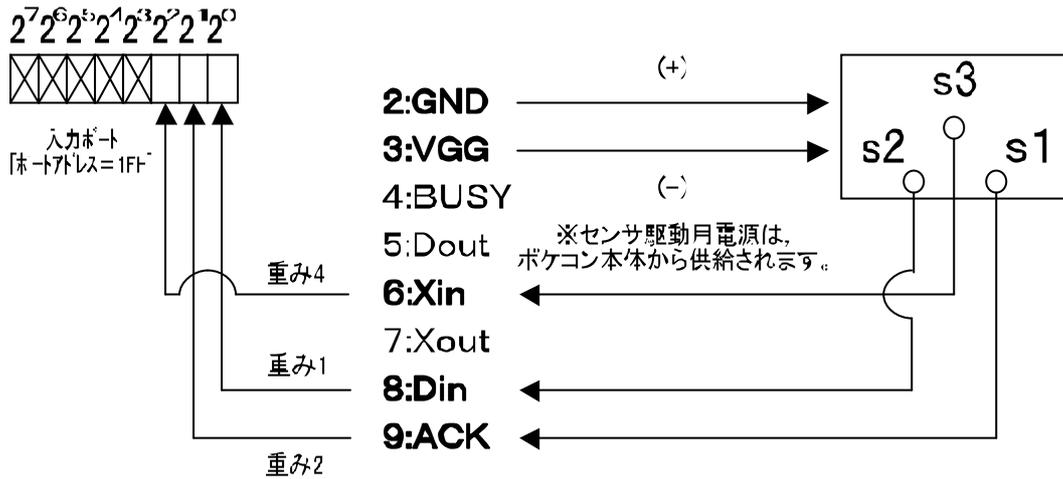
「センサ入力」の仕組み

INP命令 = BASICによる「センサ入力」



問題 左ページと次の図を参考に，下図のセンサの値を読み取ろう。答えは，
2進数と16進数で書くこと。 ヒント：pokeD を実行してみよう

IN命令 — マシン語による「センサ入力」



マシン語の「IN命令」は，入力ポート（=11pin 端子）から直接入力するため，「タイムラグ」は限りなくゼロに近づきますが，入力データの形式がBASICと異なるので注意が必要です。 [6:Xin=重み4，8:Din=重み1，9:ACK=重み2]

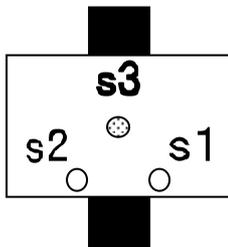
$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$	$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$	$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$	$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$
$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$	$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$	$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$	$\begin{matrix} s3 & s1 & s2 \\ (\quad \quad)_2 \\ [16 \text{ 進数}] = \end{matrix}$

(1) マシン語による「ライトレース制御」 ~ センサ(s1, s2)に注目~

<考え方> 基本的な制御方法は,【21~22ページ】の2個のセンサによる「ライトレース」制御と同じです。

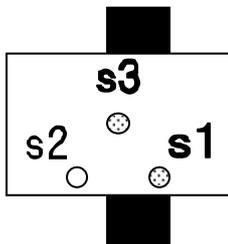
ただし, BASIC (C言語)の場合と違って, センサ s1 と s2 の重みが異なる点に注意が必要です。

問題 2個のセンサ(s1, s2)による制御の場合, 下図のセンサ入力(16進数)と, モータ出力(16進数)はどうか? ヒント: pokeE を実行してみよう



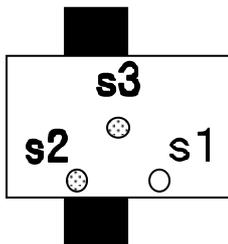
コース (ライン中央) s3 s1 s2
 センサ入力 (IN命令) = (1 0 0)₂ = _____
 [s1 = 0, s2 = 0]なので, 制御データは「直進」

モータ出力 = (OUT命令) _____



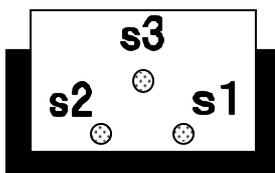
コース (左に少しずれた状態) s3 s1 s2
 センサ入力 (IN命令) = (1 1 0)₂ = _____
 [s1 = 1, s2 = 0]なので, 制御データは「右へ軌道修正」

モータ出力 = (OUT命令) _____



コース (右に少しずれた状態) s3 s1 s2
 センサ入力 (IN命令) = (1 0 1)₂ = _____
 [s1 = 0, s2 = 1]なので, 制御データは「左へ軌道修正」

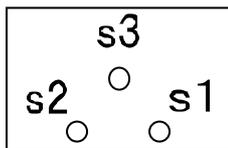
モータ出力 = (OUT命令) _____



コース (停止位置) s3 s1 s2
 センサ入力 (IN命令) = (1 1 1)₂ = _____
 [s1 = 1, s2 = 1]なので, 制御データは「ブレーキ」

モータ出力 = (OUT命令) _____

整理してまとめる	センサ入力	モータ出力	センサ入力とモータ出力の関係は BASIC (C言語) の場合と 異なるので、注意が必要です。
コース			
コース			
コース			



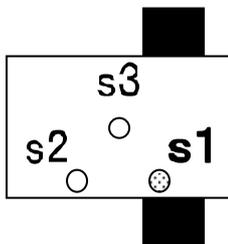
コース (ラインの外)

s3 s1 s2

センサ入力 (IN命令) = (0 0 0)₂ = _____

[s1 = 0, s2 = 0]なので、制御データは「直進」

モータ出力 = (OUT命令) _____



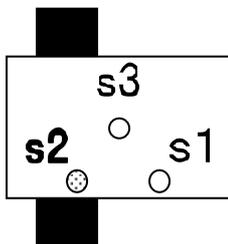
コース (左にずれた状態)

s3 s1 s2

センサ入力 (IN命令) = (0 1 0)₂ = _____

[s1 = 1, s2 = 0]なので、制御データは「右へ軌道修正」

モータ出力 = (OUT命令) _____



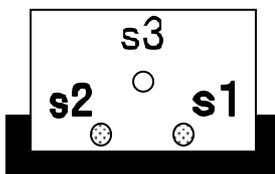
コース (右にずれた状態)

s3 s1 s2

センサ入力 (IN命令) = (0 0 1)₂ = _____

[s1 = 0, s2 = 1]なので、制御データは「左へ軌道修正」

モータ出力 = (OUT命令) _____



コース (停止位置)

s3 s1 s2

センサ入力 (IN命令) = (0 1 1)₂ = _____

[s1 = 1, s2 = 1]なので、制御データは「ブレーキ」

モータ出力 = (OUT命令) _____

整理してまとめる	センサ入力	モータ出力	センサ入力とモータ出力の関係は BASIC (C言語) の場合と 異なるので、注意が必要です。
コース			
コース			
コース			

(2) マシン語による制御プログラム

マシン語エリアの確保 (注意 : この操作を行わないとマシン語は使えません)

MON	MACHINE LANGUAGE MONITOR の起動
*USER200	マシン語エリアの確保 (FREE : 0100 - 0200)

「アセンブリ言語」のプログラムを入力し、(TEXT モードで、Edit)

「アセンブラ」でマシン語に変換する。(ASMBL モードで、Assembler)

「アセンブラ」機能が使えないポケコンは、マシン語を直接入力する。

MON	MACHINE LANGUAGE MONITOR の起動
*S110	マシン語を入力する

[マシン語] [アセンブリ言語]

0100	10; File	Name	= SET.asm	この名前で保存する。
0100	20DATA	EQU	100H	配列 DATA の先頭アドレス
0100	30CNT	EQU	108H	変数 CNT のアドレス
0100	40OFF	EQU	109H	変数 OFF のアドレス
0110	50	ORG	110H	
0110 3E08	60	LD	A, 8	ポケコンの初期設定
0112 D311	70	OUT	(11H), A	(シフトキーのチェックのために必要)
0114 3E80	80	LD	A, 80H	ポケコンの初期設定
0116 D315	90	OUT	(15H), A	(セサ入力を行うために必要)
0118 210001	100	LD	HL, DATA	HL レジスタ = 配列の先頭アドレス
011B 3A0901	110	LD	A, (OFF)	
011E 47	120	LD	B, A	Bレジスタ = 【スイッチ OFF】を設定
011F 3A0801	130LOOP:	LD	A, (CNT)	
0122 4F	140	LD	C, A	Cレジスタ = 【ループ カウント】を設定
0123 DB1F	150SINP:	IN	A, (1FH)	セサ入力
0125 E607	160	AND	7	・・・セサ入力は { 0 ~ 7 }
0127 6F	170	LD	L, A	Lレジスタ = { 配列のオフセット } を設定
0128 79	180	LD	A, C	Aレジスタ = 【ループ カウント】を設定
0129 B8	190	CP	B	Aレジスタ - Bレジスタ を実行
012A CA3601	200	JP	Z, MBRK	0 ならば, MBRK へジャンプ
012D DA3D01	210	JP	C, MOFF	負ならば, MOFF へジャンプ
0130 7E	220	LD	A, (HL)	
0131 D318	230	OUT	(18H), A	モータへ「制御データ」を出力
0133 C34101	240	JP	DISP	

0136	3E83	250MBRK:	LD	A, 83H	
0138	D318	260	OUT	(18H), A	モータへ「ブレーキ」を出力
013A	C34101	270	JP	DISP	
013D	3E00	280MOFF:	LD	A, 0	
013F	D318	290	OUT	(18H), A	モータへ「モータOFF」を出力
0141	C5	300DISP:	PUSH	BC	
0142	E5	310	PUSH	HL	
0143	7D	320	LD	A, L	Aレジスタに「セサ入力」をセット
0144	F630	330	OR	30H	文字コードに変換
0146	1603	340	LD	D, 3	表示位置(行)の設定
0148	1E17	350	LD	E, 17H	表示位置(桁)の設定
014A	CD62BE	360	CALL	0BE2H	「セサ入力」の表示ルーチン
014D	E1	370	POP	HL	
014E	C1	380	POP	BC	
014F	0D	390	DEC	C	Cレジスタ=【ループカウンタ】を-1
0150	C22301	400	JP	NZ, SINP	0 でなければ, SINP へジャンプ
0153	DB13	410	IN	A, (13H)	0 ならば, シフトキーのチェック
0155	E601	420	AND	1	・・・入力あり(1) / 入力なし(0)
0157	CA1F01	430	JP	Z, LOOP	0 なら, LOOP へジャンプ
015A	C9	440	RET		BASIC のプログラムへ戻る
015B		450	END		

問題 BASICのプログラムを完成させ、マシン語プログラムを実行しよう。

ヒント：pokeF を実行してみよう

10	' SAVE " PWM-A "	この名前で保存する。
20	POKE &H100, _____	
30	POKE &H104, _____	
40	INPUT " スイッチ ON= "; N1	
50	INPUT " スイッチ OFF= "; N2	
60	POKE &H108, N1+N2, N2	
70	PRINT " STOP ---- SHIFT "	
80	CALL &H110	マシン語プログラムの実行
90	END	

実験 1 「スイッチOFF」を1に固定して、「スイッチON」を1,2,3,...と大きくしていくと、「ポケコン搬送車」の動きはどう変わるか？

気付いたことをまとめよう。

実験 2 「スイッチON/OFF」ともに1,ともに2,ともに3,...と大きくしていくと、「ポケコン搬送車」の動きはどう変わるか？

気付いたことをまとめよう。

実験3 BASICのプログラムで「スイッチON = 10 / OFF = 10」
 の場合と、マシン語のプログラムで「スイッチON = 10 / OFF =
 10」の場合では、「ポケコン搬送車」の動きはどう変わるか？

気付いたことをまとめよう。

<一口メモ> =====

プログラムの作成と実行（マシン語編）

