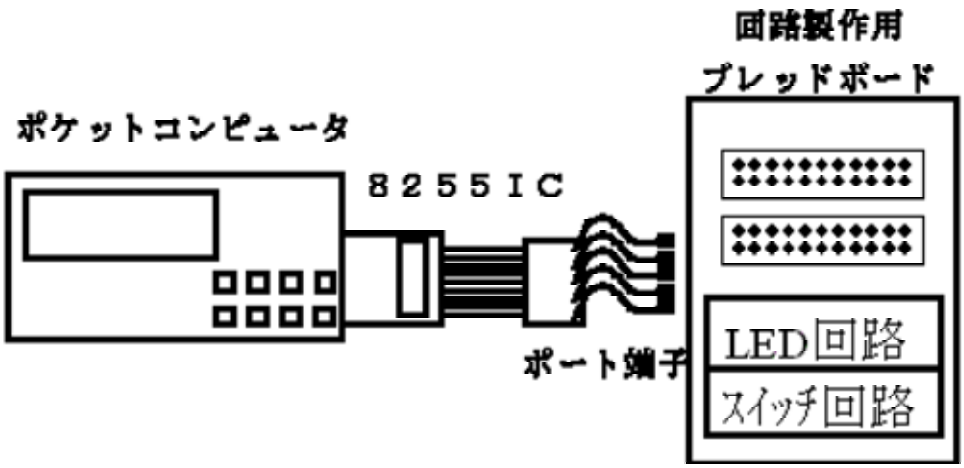
アセンブラ制御実習



愛知県立刈谷工業高等学校

機械科　　年　　組　　番

氏名

目標

１週目　進数とコンピュータの概要
　　アセンブリ言語の概要と入力法
　　簡単な転送命令

２週目　メモリーへの簡単な転送命令
　　加算・減算の基本
　　フラグレジスタについて

３週目　制御の概要とブレッドボードの使用法
　　入出力設定
　　ＬＥＤの点滅プログラム

４週目　応用プログラム
　　スイッチによる入力プログラム
　　ゼロフラグによる分岐ジャンプ

発展課題　タイマープログラムを使用したＬＥＤの点滅プログラム
　　サブルーチンの概念
　　マスクによるビットの確認と分岐ジャンプ
　　ＢＡＳＩＣ言語とのリンク（ＰＥＥＫ・ＰＯＫＥ命令）

マイクロコンピュータ制御実習

１．目的

マイクロコンピュータの簡単な仕組みと，操作法・プログラミングを学習し、データの流れや制御方法を理解する。

２．マイコンの利点

①経済性が高い（ＬＳＩ化）

部品費の減少、組立・製作費の減少，低消費電力である。

②小型・軽量化された

ＬＳＩ化により、部品数が大幅に減少。

③信頼性が高い

部品数やコネクタ等の接続点数が減少したため故障が減少する。

④融通性がある

プログラム（ソフト）の変更により、自由自在に機能を変換できる。
また、将来の拡張性がある。

⑤保守性が良い

テストプログラムの実行や故障診断プログラムによる動作チェックが容易である。

⑥標準化できる

ハードウェアや実装方法が標準化できる。ハードウェアとソフトウェアを並行して開発でき、標準的な設計法が確立しやすい。その結果設計効率が高く、設計期間も短縮できる。

３．マイコンのシステム構成

①ＣＰＵ（Central Processing Unit：中央処理装置）：脳

人間の頭脳と同じで、演算やメモリ・Ｉ／Ｏ等の動作を管理・制御するなどの、コンピュータの中心的な役割をする装置である。

②メモリ（Memory：記憶装置）：脳

メモリはデータ（命令データも含む）を記憶する所である。

・ＲＡＭ（Random Access Memory：ラム）

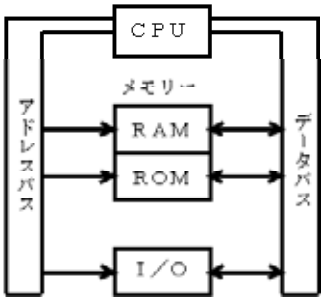
読み書き両用でデータの記憶に使用されるが電源を切ると内容は消える。

・ＲＯＭ（Read Only Memory：ロム）

読み出し専用メモリで電源を切っても内容は消えない。モニタプログラムや組み込みプログラムの記憶に利用される。

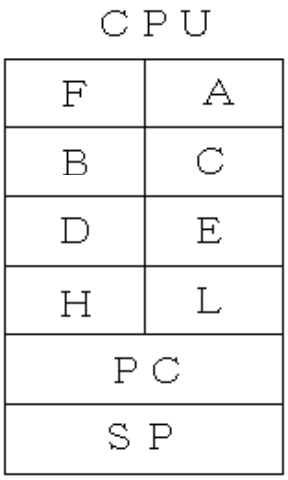
③Ｉ／Ｏ装置（Input/Output：入出力装置）：目，耳，口，手

外部からの信号をマイクロコンピュータ内部へ受け入れる役割を持っている。
また、逆にマイクロコンピュータ内部の信号を外部に出す役割もします。



④汎用レジスタ（ワーキング・レジスタ）

８ビットのA，B，C，D，E，H，Lの７つのレジスタは、ＣＰＵ内部で演算に必要なデータや番地を一時記憶させるのに用いる。よって高速に処理をすることができる。



・Ａレジスタ：アキュムレータ（Acc）とも呼ばれ最も機能の多いレジスタである。全ての演算（加算，減算，シフト，論理演算）や外部とのデータ転送もこのレジスターを使用して行われる。

・ＢＣレジスタ：ループの回数を数えたり、Ａレジスタの一時待避、あるいは定数を記憶して置くのに使用される。また、１６ビットのペアレジスタとして、メモリアドレスの働きもある。

・ＤＥレジスタ：ＢＣより用途が多いレジスタである。ＨＬペアレジスタとの間で加算ができる。

・ＨＬレジスタ：Ａレジスタに次いでよく用いるレジスタである。

⑤フラグ（Ｆ：Flag）

フラグは演算を実施した結果，設定された条件が満たされたか否を判定するのに用いられる、その結果プログラムの流れを変えることが可能である。桁上がりをしたかや計算結果がゼロであるかなど５種類のフラグがある。

４．コンピュータの概要

（１）数体系（２進数，１０進数，１６進数）

マイコンを含め、ディジタル計算機の計算は電圧が高いか低いか、電流が流れているか、いないかなどの２値信号（２進数）によって判断されている。２進数では桁数が多くなると取扱いにくいので、扱いやすいように１６進数が用いられる。

１６進数では１桁は４ビットの情報量をもつので長い２進数は最下位桁から４桁ずつ区切って個々に１６進表示すれば、それがそのまま１６進数を表す。一般に１６進数は最後にHをつけて書く。情報量として８ビット（１バイト）を用い、１６進２桁で表現する。また、２進数に変換しやすい３ビット区切りの８進数も使用される。

２進数・８進数・１６進数について

・１０進数――――――人間のわかりやすい数字
・２進数――――――――ＯＮとＯＦＦの電気信号を数字で表現している。
・１６進数――――――２進数の４ビット区切りを一桁になるように表現している
　　プログラムの入力に適している。
・８進数――――――――２進数の３ビット区切りを一桁になるように表現している。
　　どのビットが１であるか変換しやすいので制御などにもちいる。

問１　下記の進数の変換をしましょう。

①（４５）$_{10}$＝（　　）$_{2}$＝（　　）$_{8}$＝（　　）$_{16}$
②（１０１１０１１１）$_{2}$＝（　　）$_{8}$＝（　　）$_{16}$＝（　　）$_{10}$
③（ＢＣ）$_{16}$＝（　　）$_{2}$＝（　　）$_{10}$
④（３７５）$_{8}$＝（　　）$_{2}$＝（　　）$_{10}$

５．ポケコンによる機械語入力

（１）初期設定

①モニターモードにする。

BASICモードでＭＯＮ(ﾓﾝ)と入力してマシン語入力ができるモニターモードにする。

②ユーザーエリアの確保（プログラム入力用のメモリー領域を確保する。）

＊ＵＳＥＲ

ＦＲＥＥ：０１００－０２００

（現在のユーザーエリヤが０１００Ｈ番地から表示される）

＊ＵＳＥＲ０２００

と入力すると１００Ｈ番地～２００Ｈ番地まで確保される

（プログラムの長さを考えて、余裕をみて確保する）

（２）モニターモードの基本命令

①メモリーセット命令（メモリーへの数値の入力命令）

＊Ｓ１００

１００Ｈ番地の現在の内容を表示してされるので変更する数値を入力する。

変更をしないときはエンターキーを押す。

０１００：１０－３Ｅ
０１０１：１０－０１
０１０２：２０－Ｆ７
０１０３：２０－

例）下記のプログラムをメモリーの１００Ｈ番地から入力して実行してみよう。

１００：　－０６
１０１：　－０６　←→ 表示される文字数
１０２：　－１６
１０３：　－０３　←→ 表示されるＹ軸の座標
１０４：　－１Ｅ
１０５：　－０９　←→ 表示されるＸ軸の座標
１０６：　－２１
１０７：　－０Ｄ　↕ 表示文字が入っている先頭番地
１０８：　－０１
１０９：　－ＣＤ　文字を表示するシステムプログラムの実行
１０Ａ：　－Ｆ１　↕ システムプログラムの先頭番地
１０Ｂ：　－ＢＦ
１０Ｃ：　－Ｃ９　←→ モニタープログラムに戻る
１０Ｄ：　－Ｂ７
１０Ｅ：　－Ｄ０
１０Ｆ：　－ＢＤ　↕ 表示されるキャラクタ・コード
１１０：　－ＢＡ
１１１：　－ＤＥ
１１２：　－Ｂ２
１１３：

②ダンップ命令（メモリー内の数値の表示命令）

Ｄ１００

とすると、メモリーの１００Ｈ番地～１０ＦＨ番地までに入っている数値を表示する

０１００：　０６　０６　１６　０３　．　．　．　．

（０５）　１Ｅ　０９　２１　０Ｄ　．　．　！　．

０１　ＣＤ　Ｆ１　ＢＦ　．　へ　円　ソ

Ｃ９　Ｂ７　Ｄ０　ＢＤ　ノ　キ　ミ　ス

・メモリーの内容により表示内容は違います。

・次のメモリー表示はリターンキーを押すと表示される。

・抜け出るときはＢＲＥＡＫキーかＣＬＳキーを押す

③プログラム実行命令（プログラムを指定番地から実行する命令）

Ｇ１００

とすると、メモリーの１００Ｈ番地からの命令を実行する。

＜抜け出るときはＢＲＥＡＫキーを押す。＞

（３）プログラムで実験

プログラムの概要は１０ＤＨ番地から表示文字がキャラクタ・コードとして入力されており、１０１Ｈ番地には表示文字数が入力されているプログラムです。

よって君たちも表示したい文字をキャラクタ・コードを参考に１０ＤＨ番地からの数値入力して、その表示個数を１０１Ｈ番地に入力します。

実行して表示されるか確認しましょう。

右の表はローマ字を表示するキャラクタコード表です。この表の数字と文字により好きなイニシャルや名前・言葉を表示してみよう。

・表示したいローマ字

（　　）

・文字数（　　）

・キャラクタコードでの数値

（　　）

上位桁	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
16進数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ヌル		スペース	0	@	P	'	p		┴		ー	タ	ミ	＝	×
1			!	1	A	Q	a	q	▁	┬	。	ア	チ	ム	ㇳ	円
2			"	2	B	R	b	r	▂	┤	┌	イ	ツ	メ	ㇴ	年
3			#	3	C	S	c	s	▃	├	」	ウ	テ	モ	ㇵ	月
4			$	4	D	T	d	t	▄	─	、	エ	ト	ヤ	◢	日
5			%	5	E	U	e	u	▅	┼	・	オ	ナ	ユ	◣	時
6			&	6	F	V	f	v	▆	│	ヲ	カ	ニ	ヨ	◥	分
7			'	7	G	W	g	w	▇	┃	ァ	キ	ヌ	ラ	◤	秒
8			(	8	H	X	h	x	▏	┌	イ	ク	ネ	リ	♠	
9			)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル	♥	
A			*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ	◆	
B			+	;	K	[	k	{	▋	┘	ォ	サ	ヒ	ロ	♣	
C			,	<	L	¥	l	:	▉	┌	ャ	シ	フ	ワ	●	
D			−	=	M	]	m	}	▉	┐	ュ	ス	ヘ	ン	○	
E			.	>	N	^	n	~	▉	└	ョ	セ	ホ	゛	／	
F			／	?	O	_	o		＋	┘	ッ	ソ	マ	゜	＼	

＜注意事項＞

必ずモニター編集に戻るようにプログラムの最後にはＲＥＴ命令（機械語　Ｃ９）または、全レジスタの内容を表示してからモニターにもどるＲＥＴ３０Ｈ命令（機械語　Ｆ７）を使用する。一般的なプログラム終了命令のＨＡＬＴ命令（機械語　７６）などを使用するとプログラム実行後にモニター編集に戻らずキー入力を受け付けないのでＲＥＳＥＴボタンを押すことになるので注意してください。

６．アセンブリ言語

コンピュータはＯＮとＯＦＦだけの世界です。しかし、それでは人間に対してあまりにわかりづらいのでＯＮを１、０ＦＦを０とした数値として考える２進数を考え出した。
また、これを入力しやすいように１６進数表示でプログラムを入力するのが機械語（マシン語）である。しかし、その数値も人間にはただの数値の羅列になり、命令としての意味が分かりづらいものである。そこで命令の数値に人間の分かりやすい英単語の記号（ニーモニックコード）を当てはめて、プログラミングを分かりやすく表記するのをアセンブリ言語と読んでいる。アセンブリ言語で書かれたソースプログラムを機械語になおす作業をアセンブル作業と呼び、その作業を人間が手で行うことをハンドアセンブルと呼ぶ。また、コンピュータが変換プログラムを使用して機械語に変換することをアセンブラと呼ぶ。

例題をアセンブリ言語で表記すると

```
        ＯＲＧ　１００Ｈ          ；機械語を入れる開始番地を指定する
        ＬＤ　Ｂ，０６Ｈ          ；Ｂレジスターに表示する文字数を入れる
        ＬＤ　Ｄ，０３Ｈ          ；Ｄレジスターに表示位置のＹ軸を入れる
        ＬＤ　Ｅ，０９Ｈ          ；Ｅレジスターに表示位置のＸ軸を入れる
        ＬＤ　ＨＬ，ＭＯＪＩ      ；ＨＬレジスターに表示文字収納番地を入れる
ＭＯＪＩ：ＣＡＬＬ　ＢＦＦ１Ｈ   ；表示プログラムを呼び出す。
        ＲＥＴ                    ；モニターに戻る。
        ＤＢ　“キミスゴイ”        ；ＲＥＴ命令後のメモリー番地からデータを
                                  ；メモリーに収納する。
        ＥＮＤ                    ；プログラム終了
```

一般的には下図のような表のコーティング用紙にアセンブリ言語を記述します。

ラベル	ニーモニック	オペランド	アドレス	マシン語	コメント
	ＯＲＧ	１００Ｈ			開始ｱﾄﾞﾚｽ指定
	ＬＤ	Ｂ、０６Ｈ	１００番地	０６　０６	Ｂに６を入れる
	ＬＤ	Ｄ、０３Ｈ	１０２番地	１６　０３	Ｄに３を入れる
	ＲＳＴ	３０	１０４番地	Ｆ７	レジスタ群を表示
	ＲＥＴ		１０５番地	Ｃ９	モニターに戻る
	ＥＮＤ				プログラム終り

（１）ロード命令と演算命令

・いろんな数値をレジスターやメモリーに転送してみよう。

＜ＬＤ命令（ロード）：転送命令＞

問２　Ａレジスターに３２を入れて、レジスターを表示するプログラムを作り実行して確認しましょう。＜３２を１６進数にするのに注意しましょう＞

問３　メモリーの０１１０ｈ番地に５３を送るプログラムを作りましょう。
＜メモリーに直接数値を代入できないのでレジスタを介して転送する＞

問４　ＨＬのペアーレジスターを使用して、０１１０ｈ番地に４０を送るプログラムを作りましょう。

（２）加算や減算の方法

＜ＩＮＣ命令（インクリメント）：加算命令＞

・１を加算する命令である。命令数が少ないのでメモリーの節約や実行速度が速くなる。

問５　３＋１の計算をしましょう。

＜Aﾚｼﾞｽﾀｰに３を転送してAﾚｼﾞｽﾀｰを＋１する＞

問６　ＨＬペアーレジスターを使用して０１１０ｈ番地に２０を０１１１番地に３０を転送しましょう。＜ヒント：ＨＬ＝ＨＬ＋１で次の番地の指定が出来る。＞

＜ＡＤＤ命令（アッド）：数値加算命令＞

・レジスターに数値を加算する命令である。

問７　３＋４０の計算をしましょう。

＜Aﾚｼﾞｽﾀｰに３を転送してAﾚｼﾞｽﾀｰを＋４０する＞

＜ＤＥＣ命令（デクリメント）：減算命令＞

・－１をする命令である、命令数が少ないのでメモリーの節約や実行速度が速くなる。

問８　３－１を計算しましょう。

＜Aﾚｼﾞｽﾀｰに３を転送してAﾚｼﾞｽﾀｰを－１する＞

＜ＳＵＢ命令（サブトラクト）：数値減算命令＞

・レジスターに数値を加算する命令である。

問９　４０－１０を計算しましょう。

問１０　Ｂレジスターに１６を転送してその後で８減算しましょう

（３）フラグレジスター（Ｆレジスター）について

・計算するときに桁上がりをしたとか桁下がりをしたとか、計算結果げゼロになったとかを憶えておくことにより、何桁でも計算できるようにしている。このように計算したことによる状態を記憶しておくのがこのフラグレジスターである。

7	6	5	4	3	2	1	0
ｻｲﾝﾌﾗｸﾞ	ｾﾞﾛﾌﾗｸﾞ	未使用	ﾊｰﾌﾋﾞｯﾄﾌﾗｸﾞ	未使用	ﾊﾟﾘﾃｨﾎﾞﾛｰﾌﾗｸﾞ	加算・減算ﾌﾗｸﾞ	ｷｬﾘｰﾌﾗｸﾞ

＜Ｚフラグ（ゼロフラグ）＞

・演算結果がゼロのときにゼロになったことを示すために０から１になる。

問１１　Ａに１を入れて１をして、フラグの状態を確認しよう

Ｆレジスター（　　）Ｈ－－－－－－（　　）$_2$

＜Ｃフラグ（キャリーフラグ）＞

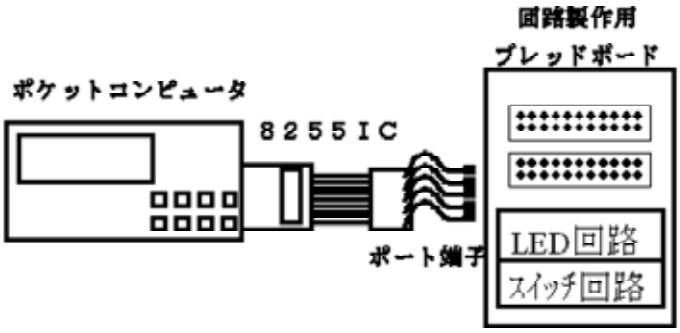
・演算結果が桁上がりしたことを示すために０から１になる。

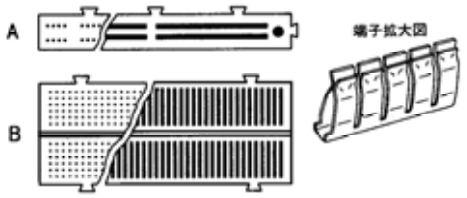
問１２　Ａに２５５をいれて＋１をしてフラグの状態を確認しよう

Ｆレジスター（　　）Ｈ－－－－－－（　　）$_2$

７．ブレッドボードとポケコンによる制御

・入出力装置としてポートを使用する。ポートとはコンピュータと外部とをつなげるためのＩＣチップであり、このＩＣで中継することによりコンピュータは安全に外部とデータのやり取りをすることにができ外部の機械を制御している。このＩＣを海と陸をつないで荷物のやり取りをする港にたとえてポートと呼ぶ。代表的なポートＩＣである８２５５ＩＣをポケコンに接続して各ポート端子を回路製作用のブレッドボードに差し込み、発光ダイオード（ＬＥＤ）の点滅や各スイッチによる入出力回路などの制御を学習する。





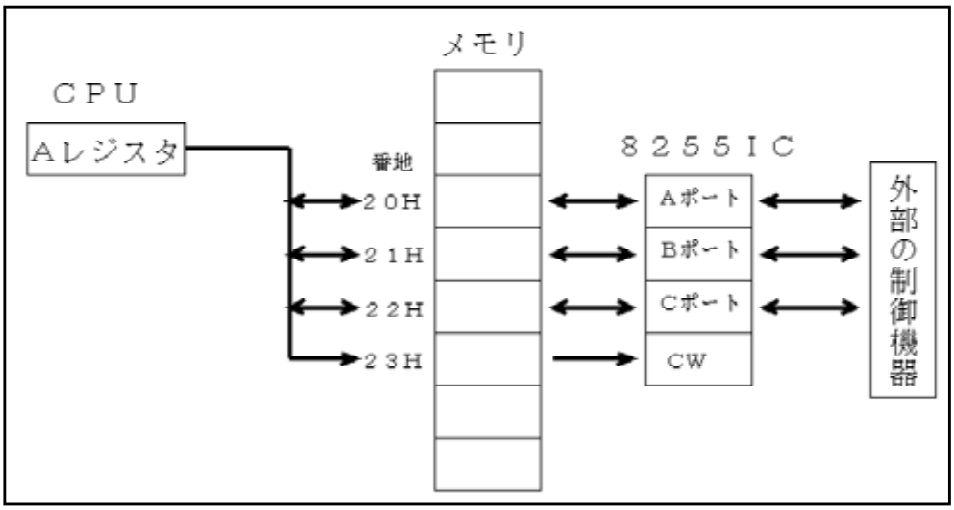
ブレッドボードは図のように
内部で接続されている。

・８２５５はプログラマブル汎用Ｉ／Ｏで、その使用形体はモード０～２までの３形体があり最も基本的なモード０について学習する。モード０では８２５５の持っている８ビット３組をプログラムによって入出力設定が出来る。

（１）入出力設定について

入出力命令は図のようにアドレスを指定する。本校ではＡポートは２０Ｈ番地、Ｂポートは２１Ｈ番地、Ｃポート２２Ｈは番地、ＣＷは２３Ｈ番地となる。

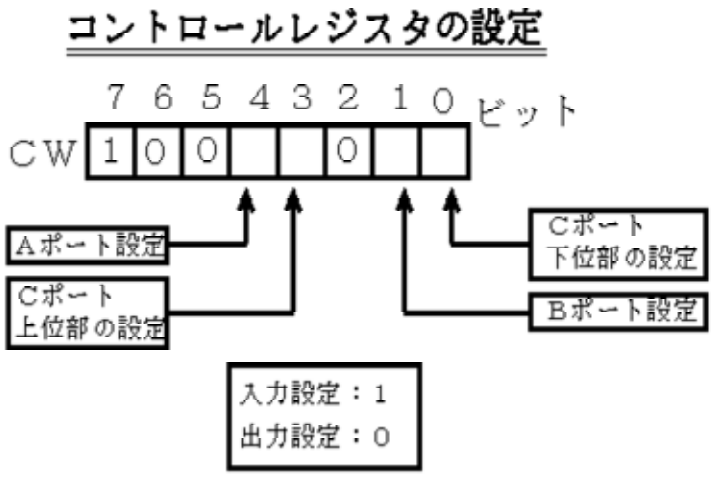
ＩＮ命令のときは指定アドレスのポートからＡレジスタに入力される。ＯＵＴ命令の時はＡレジスタの内容を指定アドレスのポートに出力する。



（２）コントロールワードとイニシャライズ（ポート設定）

各ポートを入力にするのか、出力にするのかを設定するポート設定をイニシャライズと呼ぶ。コントロールワード（ＣＷ）レジスタを操作することにより、各ポートの入出力が設定される。コントロールワードでの各ポートの入出力の設定法は下図のように指定のビットを設定することで決めることが出来る。

ＡポートとＢポートは８ビット単位で設定するが、Ｃポートは特別で上位４ビットと下位４ビット別々に入出力を設定することが出来る。



＜例＞Ａ・Ｃポート入力で，Ｂポート出力の時のコントロールワードを示す。

ＣＷ１００１１００１ ＝ ９９Ｈ

使用しないポートは入力指定にするのが望ましい。

＊入出力制御プログラムでは、指定のポートアドレスにデータを入出力する前にコントロールワードを必ずイニシャライズする。

問１３　Ｂ・Ｃポート入力，Ａポート出力のコントロールワード（ＣＷ）の値を求めましょう

ＣＷ 1 0 0 0 －－－－－－－－（　　）１６進数

問１４　出力はＡポート・入力をＢポートでＣポートは全て出力で設定するしてみよう。

ＣＷ 1 0 0 0 －－－－－－－－（　　）１６進数

問１５　ＬＥＤを全て点灯するプログラムを作りましょう。
　　ただし、出力はＡポート（２０Ｈ番地）とする

ラベル	ﾆｰﾓﾆｯｸ	オペランド	アドレス	マシン語	コメント
	ＯＲＧ	１００Ｈ			開始ｱﾄﾞﾚｽ指定
			１００番地		イニシャライズ
					Ａﾎﾟｰﾄ出力・ＢとＣﾎﾟｰﾄ入力
					Ａ＝（　　）Ｈ
					（Ａポート）＝Ａ
					レジスタ群を表示
					モニターに戻る
	ＥＮＤ				プログラム終り

問１６　ＬＥＤが下記のように点滅するプログラムを作りましょう。

7	6	5	4	3	2	1	0	
ON	OFF	ON	OFF	ON	OFF	ON	OFF	－－－（　　）１６進数

ラベル	ﾆｰﾓﾆｯｸ	オペランド	アドレス	マシン語	コメント
	ＯＲＧ	１００Ｈ			開始ｱﾄﾞﾚｽ指定
			１００番地		イニシャライズ
					Ａﾎﾟｰﾄ出力・ＢとＣﾎﾟｰﾄ入力
					Ａ＝（　　）Ｈ
					（Ａポート）＝Ａ
					レジスタ群を表示
					モニターに戻る
	ＥＮＤ				プログラム終り

問１７　スイッチの状態を入力して、レジスタに格納するプログラムを作って内容を確認しましょう。
　　ヒント：Ｂポート（２１Ｈ番地）を入力にして、そこからスイッチの状態を読み込む

問１８　実行するとスイッチの状態と同じようにＬＥＤが点灯するプログラムを作りましょう。
　　＜何度もスイッチを変更して実行し、正しく点灯するか確認しましょう＞
　　ヒント：Ｂポート（２１Ｈ番地）の内容を入力、その内容を
　　そのままＡポート（２０Ｈ番地）より出力する。

（3）分岐命令

・プログラムの流れを変えるものであり、ＢＡＳＩＣ原語のＧＯＴＯ命令やＩＦ命令・ＳＵＢ命令にあたる。

＜ＪＰ命令（ジャンプ）無条件ジャンプ＞

ＪＰ　＜指定番地＞・・・・・・・・・・・無条件で指定番地に飛ぶ

マシン語にするときには指定番地の上位と下位が入れ替わるので注意すること。

問１９　スイッチの状態と同じようにＬＥＤが点灯するのを繰り返すプログラムを作りましょう　＜プログラムの停止はBREAKキーを押して止めます。＞

ヒント：Ｂポート（２１Ｈ番地）の内容を入力、その内容をそのままＡポートより出力する。これを繰り返す。

ラベル	ﾆｰﾓﾆｯｸ	オペランド	アドレス	マシン語	コメント
	ＯＲＧ	１００Ｈ			開始ｱﾄﾞﾚｽ指定
	ＬＤ	Ａ，８０Ｈ	１００番地		イニシャライズ
	ＯＵＴ	（２３Ｈ）			Aﾎﾟｰﾄ出力・Bﾎﾟｰﾄ入力
LOOP	ＩＮ	（２１Ｈ）			Ａ＝（Ｂポート）
	ＯＵＴ	（２０Ｈ）			（Ａポート）＝Ａ
	ＪＰ	LOOP			ＧＯＴＯ　LOOP
	ＲＳＴ	３０			レジスタ群を表示
	ＲＥＴ				モニターに戻る
	ＥＮＤ				プログラム終り

＜ゼロフラグなどによる条件ジャンプ＞

ＪＰ　Ｚ，＜指定番地＞・・・・・・・・ゼロフラグが０のとき、指定番地に飛ぶ

ＪＰ　ＮＺ，＜指定先番地＞・・・・・・ゼロフラグが０でないときに指定番地に飛ぶ

問２０　スイッチで入力した値が０１Ｈのときプログラムを終了するプログラムを作りましょう。

ヒント：Ｂポートで入力した値を－１して、ゼロでなければ繰り返して、ゼロならばプログラムを終了する分岐を考える。

ラベル	ﾆｰﾓﾆｯｸ	オペランド	アドレス	マシン語	コメント
	ＯＲＧ	１００Ｈ			開始ｱﾄﾞﾚｽ指定
	ＬＤ	Ａ，８０Ｈ	１００番地		イニシャライズ
	ＯＵＴ	（２３Ｈ）			Aﾎﾟｰﾄ出力・Bﾎﾟｰﾄ入力
LOOP	ＩＮ	（２１Ｈ）			Ａ＝（Ｂポート）
	ＤＥＣ	Ａ			Ａ＝Ａ－１
	ＪＰ	NZ，LOOP			IF（NZ）THEN　LOOP
	ＲＳＴ	３０			レジスタ群を表示
	ＲＥＴ				モニターに戻る
	ＥＮＤ				プログラム終り

問２１　スイッチの状態が００００１１１１になったとき、全てのＬＥＤを点灯するプログラムを作りましょう。

＜ヒント：入力した値から０ＦＨを引いてゼロになるかを確認する＞

８．応用プログラム

（１）タイマーの原理

コンピューターが演算などの仕事をしているときに、人間から見ると停止している様に見える。この現象を利用して、一定時間停止させるように見せるプログラムである。

問２２　２５５から１を減算していき、０になったら停止するプログラムを作りましょう。

問２３　ＢＣレジスターを使用して１６ビットの演算により、２５５より停止時間が長いタイマーを制作しましょう。

問２４　時間により点灯していたＬＥＤの０ビット目が消えるプログラムを作りましょう。

（２）サブルーチンの概要

何度も使用するプログラムはサブルーチンとして制作して、何度でも呼び出せるようにしておくと便利である。

問２５　ＬＥＤの光が右から左に流れるプログラムを作りましょう。ただし、タイマープログラムをサブルーチンに変更して呼び出すプログラムを作りましょう。

ヒント：1･2･4･8･16･32･64･124 の重みで右から左に流れる様に見える。

（３）マスク命令

マスクとは各ビットで指定のビットだけを知りたいときに、他のビットは見えなくなるようにするためにすることをマスクを掛けると表現する。実際には関係のないﾋﾞｯﾄは全てゼロになるようにして、指定のビットの変化だけがあらわれるようにする。
そのためには指定ビットを１として、他のビットをゼロにした数字と論理積を取ることにより指定ビットの変化を知ることが出来る。

例）指定ビットが４ビット目の変化を知りたいとき

	7	6	5	4	3	2	1	0	
	1	0	1	1	0	1	1	1	（読み込んだデータだとすると）
ＡＮＤ	0	0	0	1	0	0	0	0	（マスクを掛けるデータ）
	0	0	0	1	0	0	0	0	

指定ビットの変化だけが表れる。

よって、指定ビットが０のときは答も０となり、１のときはゼロ以外になる。
これにより、ゼロフラグを見ることによりいろいろな条件ジャンプが使用できる。

問２６　スイッチ０（０ﾋﾞｯﾄ目）が押されたらＬＥＤをすべて点灯する

（４）ＢＡＳＩＣ言語とマシン語のリンク

ＰＥＥＫ命令：メモリーに数値を代入する。（例：ＰＯＫＥ　＆Ｈ１０５，ＦＦＨ）
ＰＯＫＥ命令：メモリーから数値を入力する。（例：Ａ＝ＰＥＥＫ　＆Ｈ１０７）
ＣＡＬＬ命令：機械語プログラムを実行する。

問２７　ＢＡＳＩＣ言語よりＬＥＤを点灯させるプログラムを作りましょう。

```
１０　ＩＮＰＵＴ　Ａ
２０　ＰＯＫＥ　＆Ｈ１０５，Ａ
３０　ＣＡＬＬ　＆Ｈ１００
４０　ＧＯＴＯ　１０
```

マシン語にて１００番地からＬＥＤの点灯プログラムの問１７を入力しておく。

問２８　タイマーを For-Next 命令で作り、ＬＥＤが右から左に点灯するプログラムを作りましょう。ヒント：1･2･4･8･16･32･64･124 の重みで右から左に流れる様に見える。