v2.02

"Yes." I replied.

"It allows a computer to talk to things outside itself, like a keyboard, mouse, hard drive, monitor, and network connection." said Pat.

"This is correct," I said "and I/O memory is also used to allow a computer to control robot motors."

Pat's mouth dropped open with surprise. "I/O memory can be used to control robot motors?" asked Pat. "How?"

v2.02

"Yes." I said. I then launched the emulator and used the Enter command to place a 00000001 binary into output port 0a200h.

```
-e a200 01
```

After entering this command, here is what was shown on the emulator's display:

"Cool!" cried Pat. "Can we turn all of the LEDs on now?"

"Sure," I said "what number do I have to pass to the Enter command in order to do this?"

Pat looked at the ceiling for a few moments then said "FF hex."

I then entered the following line into the emulator:

```
-e a200 ff
```

And here was what was shown on the display:

Pat was very excite'& y this and one could almost see the gears turning behind those bright eyes. Finally Pat said "Can we make a program that blinks all of the LEDs on and off continuously?"

"Okay." I said. I then create the following program, assemble it, loade'&t into the emulator, and execute'&t:

```
%uasm65,description=""
;Program Name: b in!"
;
```

```
)ain (
;%urn a   t&e  ig&ts on and t&en *aste some time
; so t&at t&e user can see t&e  ig&ts on"
       da +11111111b
      sta 0a200&
      ,sr de a'

;%urn a   t&e  ig&ts off and t&en *aste some time
; so t&at t&e user can see t&e  ig&ts off"
       da +00000000b
      sta 0a200&
      ,sr de a'

      ,mp )ain

;-.it t&e program"
      br!


;(((((((((((((((((((((((((((((((((((
;        /ubroutines area"
;(((((((((((((((((((((((((((((((((((

;(((((((((((((((((((((((((((((((((((
;$e a' subroutine"
;
;%&e purpose of t&is subroutine is to generate
; a de a' so t&at t&e rate of t&e b in!ing
; can be contro  ed"
;
;0&ange t&e number t&at is being  oaded into
; t&e 121 register to c&ange t&e de a' time"
;(((((((((((((((((((((((((((((((((((
$e a' (
;/a3e registers on t&e stac!"
      p&a
      t.a
      p&a
      t'a
      p&a

;P ace 'o inte t&e counr roun tim-ee  t&e counr roun tim-a
```

```
     da 0a400&
    bne 5ai t7oop%op

;8estore registers from t&e stac!"

;((((((((((((((((((((((((((((((((((((((
;        #ariab es area"
;((((((((((((((((((((((((((((((((((((((

%9uasm65
```

timer circuit wehich isg interfaced to

/cà //

```
%uasm65
;Program Name: s*2 eds"
;
;#! s sion: 1"01"
;
;$!  crip!tion: %&!  0p!urpos  0of  t&!is p!rogram is to &!a3!
; t&!  07-$!s in p!o!rt 0!a20!0&! ref  ect t&!  0st!at!  0of  t&!
; s*itc&!  "

;((((((((((((((((((((((((((((((((((((
;      Program entr' p!o!int"
;((((((((((((((((((((((((((((((((((((
      o!rg 0!20!0&. stedain (


       da 0!a60!0&. sdwst!a 0!a20!0&. sdmp )ain




      end. sp!=d0uasm65
```

v2.02

```
    10110110 - :rigina  bit patten"
2N$ 00000100 - ;it )as!"
    --------
    00000100 - 8esu t"
```

"Notice that each of the original bits that are ANDed with 0 in the bit mask are turned into 0 bits, but the bit that was ANDed with a 1 bit in the bit mask remained what it was." I said.

"Pat studied the AND operation I had just performed with a look of confusion then said "I'm still not getting how this is useful."

"Perhaps if I use AND in a program, it will make better sense to you." I then created the following program:

```
%uasm65,description=""
;Program Name: s*itc& et"
;
;#ersion: 1"01"
;
;$escription: %&e purpose of t&is program is to output a different
```

v2.02

```
0!/*6 (
       da 0a600&
      and +01000000b
      be> 0!/*G
```

[illegible]

```
t'a
p&a
```

achieve the desired result."

```
   10110010 - :rigina  bit patten"
:8 00000100 - ;it )-bt)-bt0110
```

v2.02

v2.02