WARNING: Some of the switches on this unit correspond with some of the sensors on the robot. Therefore operation of this unit may prevent the robot from calibrating. Always CALIBRATE the robot before connecting this unit.

The I/O simulator is a simple box with 8 LED's and 5 switches.
The LEDs are connected to the PA output port via the 15 way D connector on the controller rear panel.

The LEDs represent bits 7 to 0, left to right i.e. the least significant digit is furthest right (just as with Decimal).

The switches are connected to the PB input port via the 9 way D connector on the controller rear panel. A slave connector is provided so your real world devices can still be connected. The switches are connected to bits 7,6,5,4, left to right.

Once you have the unit connected make sure all the switches are in the up position. Note that if the PB 4 switch is in the down position the robot will not calibrate.

Type START if you have not already done so.
Example definitions follow that you can enter into a project ED2 window and download or simply enter then directly into the communications window.

Outputs
Type GRIP you will see PA 0 light.
UNGRIP and it goes out.

For an electric gripper for GRIP PA 0 will be dim then stay bright and for UNGRIP PA 0 goes out and PA 1 is dim then goes out.

Type PA 1 ON – PA 1 lights
PA 2 ON – now PA 1 and 2 are both lit.
PA 1 OFF – the PA 1 light goes off leaving the PA 2 still on.

You can incorporate outputs in a definition that also has robot motion, for example let's assume LED 5 means "warning, robot in motion". You could have connected that output to an illuminated legend with that wording.
For example

```
: INITIALIZE
PA 5 ON
CALIBRATE
HOME
PA 5 OFF
;
```

Type `INITIALIZE` to observe that.

Of course you can also simply write a value to the PA port. For example

```
5 PA OUT
```

will turn on LEDs 0 and 2
(LED 0 has value 1, LED1 value 2, LED2 value 4, LED3 value 8 etc.)

Inputs

Type PP

This displays a row of 1s and 0s. With the robot at home position you would expect to see 11111111 for R17 or 11111110 for R12. We'll call that 1111111x

Now press switch PB 7. You should see

0111111x

Press PB 5

0101111x

Release PB 7

1101111x

etc.

You can make robot action dependent on the inputs. Suppose you have PB 7 connected to an operators console to a big green button marked "GO" (from operator's point of view this would start a cycle)

Try this:

```
: TEST2
CALIBRATE
PB 7 0 WAIT
HOME
;
```

type TEST2

The robot will calibrate then wait for switch PB 7 to be pressed, then go home.

Warning: switch PB4 must be OFF for the robot to calibrate (i.e. must show a 1).

In a normal industrial application the switch would have been released before the next cycle, but in some circumstances we need to wait for a signal to return to a 1 before we proceed.

```
: TEST3
CALIBRATE
PB 7 0 WAIT
PB 7 1 WAIT
HOME
;
```

type TEST3

Now the switch must be pressed and then returned to off for the robot to go HOME.

If you want the switches to determine which of 2 tasks the robot must do, then you simply want to read a switch and not wait for it. Suppose you have two PLACEs P1 and P2. Switch 7 up sends robot to P1 and switch 7 down to P2.

```
: TEST4
HOME
PB 7 BIT? IF
    P1
ELSE
    P2
THEN
;
```

type TEST4 to try it.

Inputs and outputs

Try this:

```
: TEST5
BEGIN
   PB IN
   PA OUT
?TERMINAL UNTIL
;
TEST5
```

This simply copies the state of the inputs directly to the outputs. Each switch will therefore control it's corresponding LED.

Press escape to get out of it.

Type `DE-ENERGIZE` then `TEST5`

Now you can move the robot by hand and the state of the sensors will be copied to the LEDs.

If you have a gripper connected then the waist sensor will appear to control the gripper.