

Inputs and outputs

The goal of this chapter is to show you how a microcontroller knows that one of the keys, wired to the port, was pressed.

As we already learned in this manual, the microcontroller's ports can be either inputs or outputs. The program in the microcontroller determines this. So, how do we tell the microcontroller about the keys on its ports P3.2 and P3.3? Piece of cake! Let's have a look at the example **Light2.bas**:

```
**'an example, traveling light goes off from the first on to the last one**
**'UP = P3.2, DOWN = P3.3, up/down keys**

Dim Lights As Byte , Up As Bit , Down As Bit

Lights = 1           'set initial value of the Lights variable
P1 = Lights          'Lights are on P1

Do                  'start of the DO-LOOP loop
Up = P3.2           'Up key is on the P3.2 port
Down = P3.3         'Down key is on the P3.3 port
```

Command **"Up = P3.2"** signals the microcontroller that port P3.2 "mirrors into a variable named **Up**". We can use this variable just like any other **Bit** type variable. One of the commands is the so-called **IF...THEN** statement, it checks the **Up** variable. This is how microcontroller operates: when the CPU encounters

```
If Up = 0 Then      'if Up = 0 rotate our lights
```

it compares the variable with zero **Up = 0**. If they're not equal, it continues execution at the end of **IF...THEN** statement, marked by **END IF**. Hmm, we're using two **END IF** statements in our example. Which one will the CPU recognize? According to the rule, only the first **END IF** statement, encountered during execution of the program, will be recognized. This is how the program from our example works: if the CPU detects **Up = 0**, it jumps into a loop, beginning with the **Rotate Lights , Left**.