

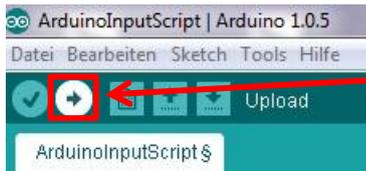


2013-29-11

1. Arduino script (ArduinoInputScript.ino)

With the help of this script you can set the digital pins on your Arduino-board on INPUT, so that you can register whether a button is pressed or not. You can upload this script to your board with the free IDE from Arduino (Arduino 1.0.5 or higher, <http://arduino.cc/en/Main/Software>).

1.1 Load the script on the board



With this button you can load the script on the board.

1.2 Code explanations to the script

```
static int baudrate = 9600;
```

This code line settles the baudrate. For another baudrate simply change the number and load the script on the Board (see the picture at 1.1). This one and the baudrate of the C#-handler must be identical.

```
static char splitter = '#';
```

Here is the dividing symbol fixed to separate the different command parts. The rhomb # was chosen to see an unequivocal separation of the button status and the pin number. This symbol can be adapted accordingly, however it must be the same as the splitter in the C#-handler.

```
for (int thisPin = 2; thisPin < 13; thisPin++)  
{  
    pinMode(thisPin, INPUT);  
    PinArray[thisPin-2] = thisPin;  
}
```

This loop is responsible for setting the pins on input. The first number (2) in the first line is the start value, the second number (13) the demolition condition. This means that now all pins from second up to the twelfth are set as INPUT. These figures can be adapted accordingly.

```
const int ledPIN = 13;  
pinMode(ledPIN, OUTPUT);
```

The first line gives a variable the value of the pin where the status-led is connected (in this case the board internal LED which is linked to the digital pin 13). The second line places the value declared in the variable on OUTPUT, so that one can see the status of the button on the board and if necessary with the help of a connected LED to pin 13 (not shining = not pressed, shining = pressed).



2013-29-11

2. Arduino Visbee Handler (ArduinoHandler.cs)

This handler enables a connection with the Arduino-Board and allows you to read the serial port.

2.1 Code-Erläuterungen zum Handler

```
private static string _portname = "COM4";  
private static int _baudrate = 9600;  
private static char _splitter = '#';
```

The first line places the port name on which the Arduino board is connected. The second one sets the baudrate. This one and the baudrate of the Arduino handler must be identical. In the third line the dividing symbol which is necessary for the unequivocal identification of the command parts is set. Also this symbol and that of the Arduino script must be identical.

```
_serialPort = new SerialPort(_portname, _baudrate);
```

Here is a serial port created with the name and the baudrate set on top. Here one must change nothing.

```
Variables.InputText.SetValue(input);
```

With this code line the values of the pin and the button status are written in the Visbee variable "InputText", which is shown in the demo application.