

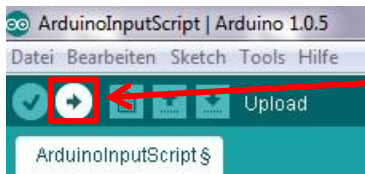


2013-05-11

1. Arduino-Skript (ArduinoInputScript.ino)

Mit Hilfe dieses Skriptes können Sie die digitalen Pins Ihres Arduino-Boards auf INPUT setzen, sodass Sie anzeigen können, ob ein Button gedrückt ist oder nicht. Dieses Skript können Sie mit der kostenlosen IDE von Arduino (Arduino 1.0.5 oder höher) auf Ihr Board laden (<http://arduino.cc/en/Main/Software>).

1.1 Upload des Skripts auf das Board



Mit diesem Button können Sie das Skript auf das Board laden.

1.2 Code-Erklärungen zum Skript

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

Diese Code-Zeile setzt die Baudrate fest. Für eine andere Baudrate einfach die Zahl ändern und das Skript auf das Board laden (siehe das Bild oben). Diese und die Baudrate des C#-Handlers müssen gleich sein.

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

Hiermit wird das Trennsymbol festgelegt, um die verschiedenen Command-Teile zu trennen. Die Raute # wurde gewählt, um eine eindeutige Trennung des Button-Status und des Pins zu sehen. Dieses Zeichen kann entsprechend angepasst werden, muss aber mit dem Splitter im Handler übereinstimmen.

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

Diese Schleife ist für das Setzen der Pins zuständig. Die erste Zahl (2) in der ersten Zeile ist der Startwert, die zweite Zahl (13) die Abbruchbedingung. Das heißt, dass jetzt nacheinander alle Pins von zweiten bis zum zwölften als INPUT gesetzt werden. Diese Zahlen entsprechend anpassen.

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

Die erste Zeile gibt einer Variablen den Wert, an welchem Pin die Status-LED angeschlossen ist (in diesem Fall die boardinterne LED an den digitalen Pin 13 gekoppelt. Die zweite Zeile setzt den in der Variablen angegebenen Wert auf OUTPUT, sodass man auf dem Board und ggf. bei einer angeschlossenen LED am Pin 13 den Status des Buttons sehen kann (nicht leuchten = nicht gedrückt, leuchten = gedrückt).



2013-05-11

2. Arduino-Visbee-Handler (ArduinoHandler.cs)

Dieser Handler ermöglicht es, eine Verbindung zum Arduino-Board herzustellen und dessen seriellen Port zu lesen.

2.1 Code-Erläuterungen zum Handler

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

Die erste Zeile setzt den Portnamen, an welchem das Arduino-Board angeschlossen ist. Die zweite legt die Baudrate fest. Diese und die im Arduino-Skript müssen die gleiche sein. In der dritten Zeile wird das Trennsymbol festgelegt, welches für die eindeutige Identifizierung der Command-Teile nötig ist. Auch dieses Zeichen und das des Skripts müssen übereinstimmen.

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

Hiermit wird ein Serial-Port erstellt mit dem oben festgelegten Namen und der Baudrate. Hier muss man nichts verändern.

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

Mit dieser Zeile wird die entsprechende Ausgabe bezüglich des Pins und des Button-Status in die Visbee-Variable „InputText“ geschrieben, welche dann in der Testapplikation angezeigt wird.