Table of contents

- **Arduino Programming Envirement and Library**
- **Switch from original Library to new Library**
- **Structure of Arduino Program for EVShield**
- **UART Example of Arduino Program for EVShield**
- **Mindsensors sensor Example of Arduino Program for EVShield**

Arduino Programming Environment and Library

If you have not already, please refer to EVShield User Guide for download instructions of Arduino Programming Environment and Library, at following url:
https://github.com/ArjanSchoorl

Switch from original Library to new Library

It is very important to know how to switch from the original Library to the newer Library. For a new user that haven't made any projects with the original Library you can skip this section and learn the structure of an Arduino Program.

First of all, the library has been simplified. That means that variables have changed and that small libraries are combined. In the examples below you can see the differences of the original Library and the new one.

The changes have been marked in <mark>yellow</mark>. You can see that the library from NXTTouch changed to EVLib. You have now only import one library if you are going to use more Lego sensors. For Mindsensors/HiTechnic sensors you have to import the MindsensorsLib.h.

For the simplicity the variable of a sensor has been changed. Every variable with EVs_ before it has changed to the variable without EVs_. You can see this in the example. So is EVs_NXTTouch changed to NXTTouch. This also counts for SH_ in the variable SH_HardwareI2C.

It's pretty easy to take the switch. If you got in trouble you can create an issue on the github page. Please don't contact me by email. ☺

```
#include <EVShield.h>
#include <EVs_NXTTouch.h>
#include <Wire.h>

// Create variables to use in this program
EVShield     evshield(0x34,0x36);
EVs_NXTTouch Touch;

void setup(){
    // start serial for output
    Serial.begin(115200);

    evshield.init(SH_HardwareI2C);
    Touch.init( &evshield, SH_BAS1 );

    Serial.println("setup done");
    Serial.println("press the touch sensor to
see changes in the values");
```

```
#include <EVShield.h>
#include <EVLib.h>
#include <Wire.h>

// Create variables to use in this program
EVShield     evshield(0x34,0x36);
NXTTouch     Touch;

void setup() {
  // Start Serial for output
  Serial.begin(115200);

  evshield.init(HardwareI2C);
  Touch.init(&evshield, BAS1);

  Serial.println("Setup done");
  Serial.println("Press the touch sensor to
see changes in the values");
```

```
    // Wait until the Go button has been
pressed
    Serial.println("Press Go button");
    evshield.waitForButtonPress(BTN_GO);
}
void loop(){
    // Create variable(s)
    int val;

    // Get the values
    val = Touch.isPressed();

    // Print the sensor values
    Serial.print("Touched? ");
Serial.println(val);
    delay(1000);

}
```

```
    // Wait until the Go button has been
pressed
  Serial.println("Press Go button");
  evshield.waitForButtonPress(BTN_GO);
}
void loop() {
  // Create variable(s)
  int val;

  // Get the values
  val = Touch.isPressed();

  // Print the sensor values
  Serial.print("Touched?:");
Serial.println(val);
  delay(1000);
}
```

Structure of Arduino Program for EVShield

In the example program below, a NXT Touch sensor and Light sensor is attached to the EVShield. Each time the touch sensor is pressed, the active/passive mode of light sensor is toggled.

The program also takes readings from light sensor continuously. You can find this program in the library distribution for EVShield. This is a great example how to use multiple Lego sensors. If you used the original library, you had to import multiple libraries. Now you have a clean overview what you use. The variables are much easier to remember and to use.

To use this program, you have to open the serial monitor. You can do this under Tools → Serial Monitor. If you have installed correctly you get the message 'Press Go button'.

```cpp
// Add all required libraries
#include <EVShield.h>
#include <EVLib.h>
#include <Wire.h>

// Create variables to use in this program
EVShield    evshield(0x34,0x36);
NXTTouch    Touch;
NXTLight    Light;

void setup() {
  // Start Serial for output
  Serial.begin(115200);

  // Initialize the shield i2c interface
  // And initialize the sensor(s) and indicate where it is connected
  evshield.init(HardwareI2C);
  Touch.init(&evshield, BAS1);
  Light.init(&evshield, BAS2);

  Serial.println("Setup done");
  Serial.println("Press the touch sensor to see changes in the values");

  // Wait until the Go button has been pressed
  Serial.println("Press Go button");
  evshield.waitForButtonPress(BTN_GO);
}
```

```
void loop() {
  // Create variable(s)
  int touchPressed;
  int lightval;

  // Get the values
  touchPressed = Touch.isPressed();
  lightval = Light.readRaw();

  // Check if touchsensor is pressed
  if (touchPressed == true){
      Serial.println("Changing light sensor to reflected light mode");
      Light.setReflected();
  }
  else{
      Serial.println("Changing light sensor to ambient light mode");
      Light.setAmbient();
  }

  // Print the value of the light sensor
  Serial.print("Light value:"); Serial.println(lightval);
  delay(1000);
}
```

UART Example of Arduino Program for EVShield

In this example program, an EV3 Ultrasonic sensor is attached to the EVShield. As you can see this library uses different modes. You can select three modes; cm, inches & Presence.

To use this program, you have to open the serial monitor. You can do this under Tools → Serial Monitor. If you have installed correctly you get the message 'Press Go button'.

```cpp
// Add all required libraries
#include <EVShield.h>
#include <EVLib.h>
#include <Wire.h>

// Create variables to use in this program
EVShield     evshield(0x34,0x36);
EV3Ultrasonic US;

void setup() {
  // Start Serial for output
  Serial.begin(115200);

  // Initialize the shield i2c interface
  // And initialize the sensor(s) and indicate where it is connected
  evshield.init(HardwareI2C);
  US.init(&evshield, BAS1);
  US.setMode(MODE_Sonar_CM);

  Serial.println("Setup done");
  Serial.println("Move object back and forth in front of ultrasonic sensor");

  // Wait until the Go button has been pressed
  Serial.println("Press Go button");
  evshield.waitForButtonPress(BTN_GO);
}

void loop() {
  // Create variable(s)
  int val;

  // Get the values
  val = US.getDist();

  // Print the sensor values
  Serial.print("Distance in cm: "); Serial.println(val);
  delay(1000);
}
```

Mindsensors sensor Example of Arduino Program for EVShield

In this example program, an Mindsensors sensor is attached to the EVShield. This example is not that much To use this program, you have to open the serial monitor. You can do this under Tools → Serial Monitor. If you have installed correctly you get the message 'Press Go button'.

```cpp
// Add all required libraries
#include <EVShield.h>
#include <MindsensorsLib.h>
#include <Wire.h>

// Create variables to use in this program
EVShield evshield(0x34, 0x36);
NumericPad myNP(0xB3);

void setup()
{
  // Start Serial for output
  Serial.begin(115200);

  // Initialize the shield i2c interface
  // And initialize the sensor(s) and indicate where it is connected
  evshield.init(HardwareI2C);
  myNP.init(&evshield, BAS1);
  myNP.InitializeKeypad();

  // Setup Code

  // Wait until the Go button has been pressed
  Serial.println("Press Go button");
  evshield.waitForButtonPress(BTN_GO);
}

void loop()
{
  // Create variable(s)
  int keys;
  keys = myNP.GetKeysPressed();

  // Loop Code
  Serial.print("Keys Pressed: ");
  Serial.println(keys);
  delay(1000);
}
```