



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:

http://www.mindsensors.com/index.php?controller=attachment&id_attachment=134

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.

Refer to comments below for explanations of parts of the program.

```
#include <Wire.h>
#include <EVShield.h>
#include <EVs_NXTTouch.h>
#include <EVs_NXTLight.h>
//
// declare the EVShield(s) attached to your Arduino.
//
EVShield evshield;

//
// declare analog devices attached to EVShield(s).
//
EVs_NXTTouch touch1;
NXTLight light1;

void setup()
{
    Serial.begin(115200); // start serial for output
    delay(500); // wait, allowing time to
                // activate the serial monitor
    Serial.println ("Initializing the devices ...");
    //

```

```

// Initialize the protocol for EVShield
// Use Hardware I2C.
//
evshield.init( SH_HardwareI2C );

//
// Wait until user presses GO button to continue the program
//
Serial.println ("Press GO button to continue");
evshield.waitForButtonPress(BTN_GO);
//
// initialize the analog sensors.
// NXT light sensor is not supported on BAS1.
// connect a NXT light sensor to BAS2,
// connect a touch sensor to BAS1
//
touch1.init( &evshield, SH_BAS1 );
light1.init( &evshield, SH_BAS2 );

bool lastTouch, touchPressed;

void loop()
{
    char    str[256];
    int lightReading;

    Serial.println("Into loop -----");

    touchPressed = touch1.isPressed();
    sprintf (str, "touch1: is pressed : %s", touchPressed?"true":"false");
    Serial.println(str);
    if ( touchPressed != lastTouch ) {
        if ( touchPressed == true ) {
            Serial.println( "Changing light sensor to reflected light mode" );
            light1.setActive();
        } else {
            Serial.println( "Changing light sensor to ambient light mode" );
            light1.setPassive();
        }
        lastTouch = touchPressed;
    }
    lightReading = light1.readRaw();
    sprintf (str, "Light sensor Reading: %d", lightReading);
    Serial.println (str);
    delay (500);
}

```

UART Example of Arduino Program for EVShield

In this example program, an EV3 Ultrasonic sensor is attached to the EVShield. The program reads the sensor information and displays it in Serial window.

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

#include <EVs_EV3Ultrasonic.h>

//
// declare the EVShield(s) attached to your Arduino.
//
EVShield    evshield(0x34,0x36);

//
// declare the i2c devices used on EVShield(s).
//
EV3Ultrasonic us1;

void setup()
{
    char str[50];

    Serial.begin(115200); // start serial for output
    delay(500); // wait, allowing time to activate the serial monitor

    Serial.println ("-----");
    Serial.println ("Starting EV3 Ultrasonic Sensor Test program");
    Serial.println ("-----");
    //
    // Initialize the protocol for EVShield
    // Use Hardware I2C.
    //
    evshield.init( SH_HardwareI2C );

    //
    // Wait until user presses GO button to continue the program
    //
    Serial.println ("Press GO button to continue");
    evshield.waitForButtonPress(BTN_GO);

    //
    // Initialize the i2c sensors.
    //
    us1.init( &evshield, SH_BAS2 );
    setMode(MODE_Sonar_CM);
}
```

```

void loop()
{
  uint8_t mode;
  float distance;
  char str[50];
  int i;

  //
  // get the reading from sensor
  //
  distance = us1.getDist();
  mode = us1.getMode();
  //
  // print the sensor staus
  //
  char *s;
  switch (mode) {
    case MODE_Sonar_Inches:
      s = "Inches";
      break;
    case MODE_Sonar_CM:
      s = "cm";
      break;
  }

  Serial.print("Distance to obstacle: "); Serial.print(distance);
  Serial.print (" (");
  Serial.print (s);
  Serial.println (");
  //
  // wait for some time
  //
  delay(1000);
}

```

i2c Example of Arduino Program for EVShield

In this example program, a NXCCurrentMeter is attached to the EVShield. The program reads the sensor information and displays it in Serial window.

```

#include <Wire.h>
#include <EVShield.h>
#include <EVs_NXCCurrentMeter.h>

//
// declare the EVShield(s) attached to your Arduino.
//
EVShield    evshield(0x34,0x36);

```

```

//
// declare the i2c devices used on EVShield(s).
//
#include <CurrentMeter.h>
void setup()
{
  char str[256];

  Serial.begin(115200); // start serial for output
  delay(500); // wait, allowing time to activate the serial monitor

  //
  // Initialize the protocol for EVShield
  // Use Hardware I2C.
  //
  evshield.init( SH_HardwareI2C );

  //
  // Wait until user presses GO button to continue the program
  //
  Serial.println( "Press GO button to continue" );
  evshield.waitForButtonPress(BTN_GO);

  //
  // Initialize the i2c sensors.
  //
  im.init( &evshield, SH_BAS1 );
}

void loop()
{
  char aa[80];
  char str[256];
  int acurr, rcurr, refl;

  strcpy(aa, im.getFirmwareVersion());
  sprintf(str, "IMeter: FirmwareVersion: %s", aa);
  Serial.println(str);

  strcpy(aa, im.getDeviceID());
  sprintf(str, "IMeter: DeviceID: %s", aa);
  Serial.println(str);

  strcpy(aa, im.getVendorID());
  sprintf(str, "IMeter: VendorID: %s", aa);
  Serial.println(str);

  /** Displays Absolute Current value
  */
  acurr = im.getACurrent();
  sprintf(str, "IMeter: Absolute Current: %d", acurr);
  Serial.println(str);
  /** Displays Relative Current value
  */

```

```
rcurr = im.getRCurrent();  
sprintf(str, "IMeter: Relative Current: %d", rcurr);  
Serial.println(str);  
/** Displays Reference Current value  
*/  
refI = im.getReference();  
sprintf(str, "IMeter: Reference Current: %d", refI);  
Serial.println(str);  
  
Serial.println( "-----" );  
delay (1500);  
}
```