User Guide for Arduino Electronic Bricks Advanced Kit v1.0

By Maker Studio

Introductions

What is Electronic Brick?

Electronic Brick (EB in abbreviation) has integrated electronic components, with unified pin header interface and with specified function. They are like the bricks used for building a house, as you can build your project easily from the ground without bothering with the messing bread board circuits. Concentrate your energy more in your idea, start from Arduino Electronic Bricks.

What kind of customers the Electronic Bricks can serve for?

ØI think QuickStartArduino

ØI want to quickly realize their creative

ØI will not weld

ØI think it is very cumbersome breadboard circuit structures

ØI do not have electronic professional background

ØI was a student

Contents


Arduino UNO R3 Clone
Electronic Brick Shield
EB-I2C LCD
EB-RGB LED
EB-Color LED
EB-LED Button
EB-Button
EB-Momentary Button
EB-5V Relay
EB-Buzzer
EB-I2C RTC
EB-Water Sensor
EB-Humidity&Temp Sensor
EB-Light Sensor
EB-Tilt Sensor
EB-Ultrasonic Sensor
EB-Infrared Sensor
EB-PIR Motion Sensor
EB-9g Servo
EB-IR Remote Controller
EB-RF Transmitter
40P Female Dupont Wires
9V Battery Connector
EB-RF Receiver
A-B USB Cable
Box
Maker Studio

Install Arduino IDE

Getting started from here: http://arduino.cc/en/Guide/HomePage

Download and Install Arduino Library for the Kit

Download the Library： http://pan.baidu.com/share/link?shareid=486204993&uk=372711376

Unzip to xxx\arduino-1.0.x\libraries, back-up and remove the LiquidCrystal library. (The LiquidCrystal lib has been integrated to the Kit library)

Experiment 01 - Blinking LED

The First experiment gets you know the development flow of Arduino projects. You may learn the following functions. (Explore more Arduino functions here: http://arduino.cc/en/Tutorial/HomePage)

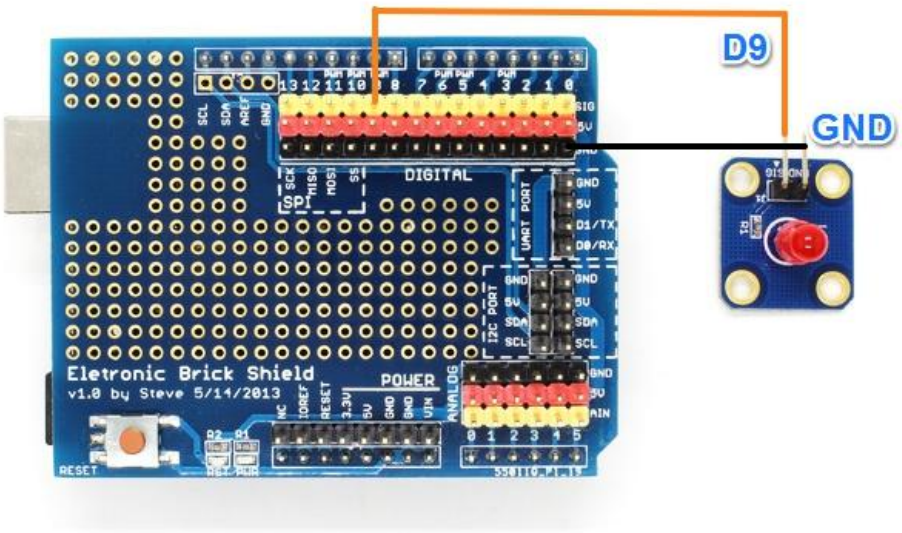
pinMode()

digitalWrite()

delay()

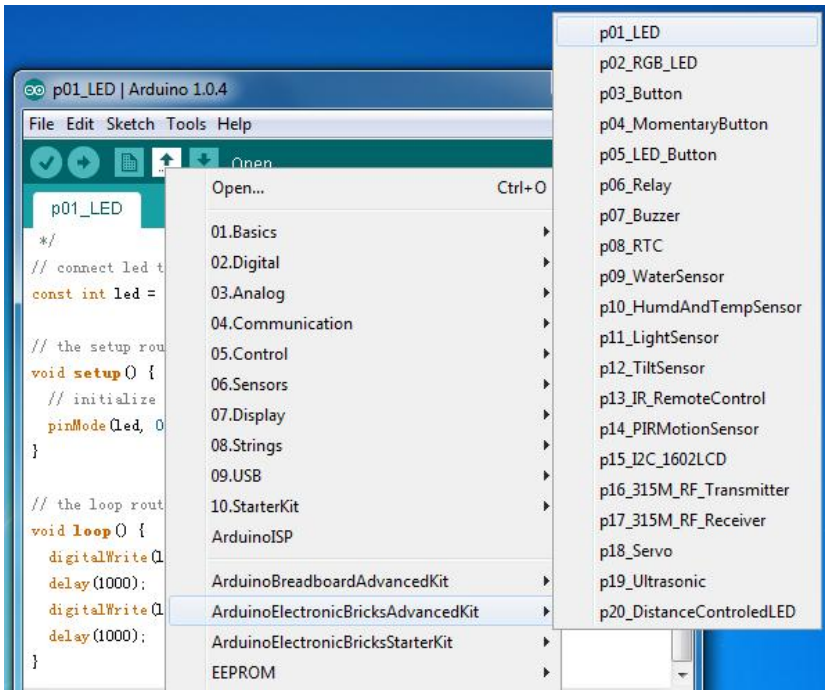
Hardware Connection

Plug the Arduino Electronic Brick Shield onto the Arduino Board. And then connect the EB – LED to the Shield based on the following picture.



Open the Arduino IDE

Click Open from the menu, and chose ArduinoElectronicBricksAdvancedKit ->p01_LED

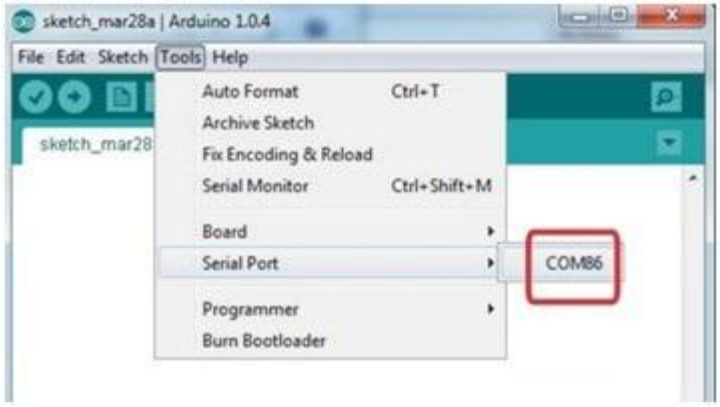


Arduino Code of p01_LED.ino

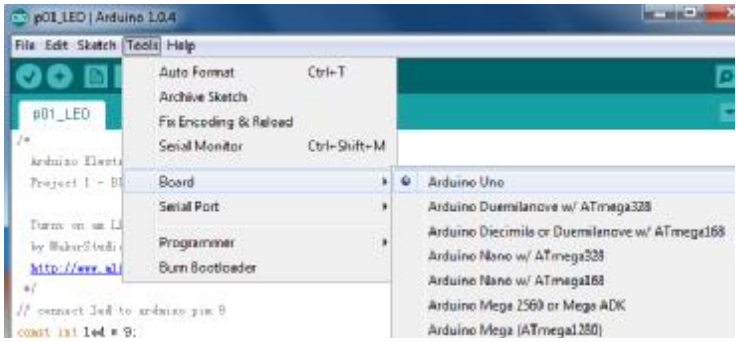
p01_LED §

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 1 - Blink LED

  Turns on an LED for one second, then off for one second, repeatedly.

  http://aliexpress.com/store/226959
 */
// connect led to arduino pin 9
const int led = 9;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

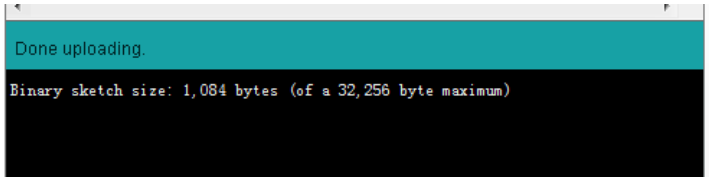
Click Tools->SerialPort, and choose the right com port, which will list after the driver successfully installed



Click Tools->Board, then choose Arduino Uno



Click Upload, the code will be compiled and uploaded to the board and shows the following information if it is success.

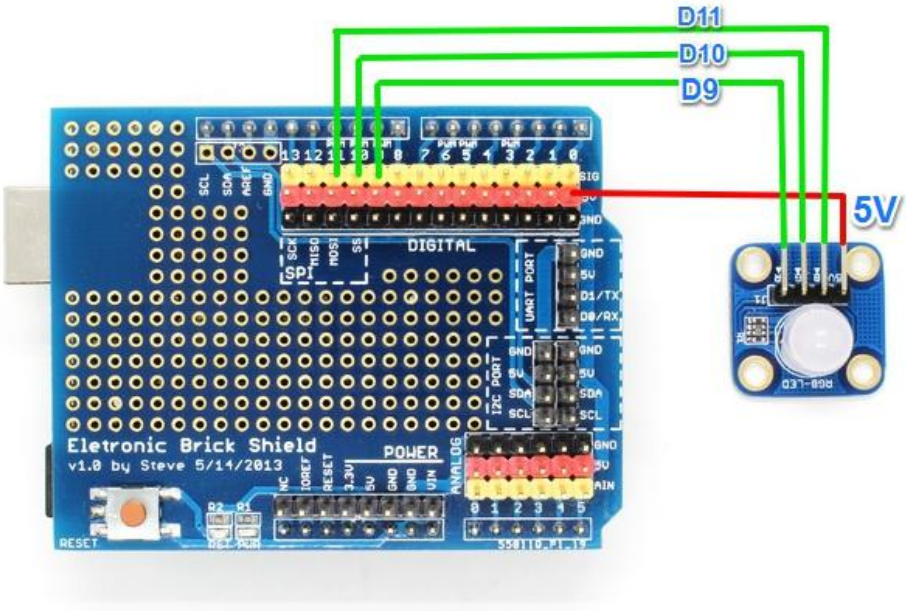


The LED of the EB will be blinking every one second.

Experiment 02 The Fading RGB LED

Learn the function of analogWrite() by controlling a fading RGB LED

Hardware Connection



Open the Arduino Code p02_RGB_LED.ino

p02_RGB_LED

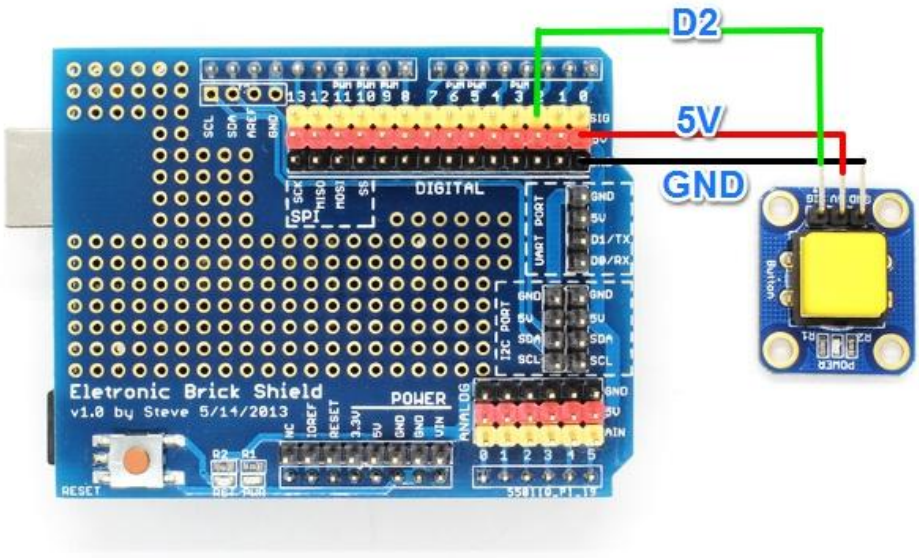
```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 2 - RGB LED

  Change the RGB LED Color and light intensity repeatedly.

  http://aliexpress.com/store/226959
 */
// connect led to arduino pin 9,10,11
const int pinR = 9;
const int pinG = 10;
const int pinB = 11;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(pinR, OUTPUT);
  pinMode(pinG, OUTPUT);
  pinMode(pinB, OUTPUT);
  digitalWrite(pinR, HIGH);
  digitalWrite(pinG, HIGH);
  digitalWrite(pinB, HIGH);
}

// the loop routine runs over and over again forever:
void loop() {
fading(pinR);
fading(pinG);
fading(pinB);
}
void fading(int ledPin)  {
  // fade out from max to min in increments of 5 points:
  for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade in from min to max in increments of 5 points:
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

When uploading success, the led will fade from red to green, and to blue.

Experiment 03 - Read the Button Input

Learn the function of digitalRead() by playing with EB – Button.

Hardware Connection



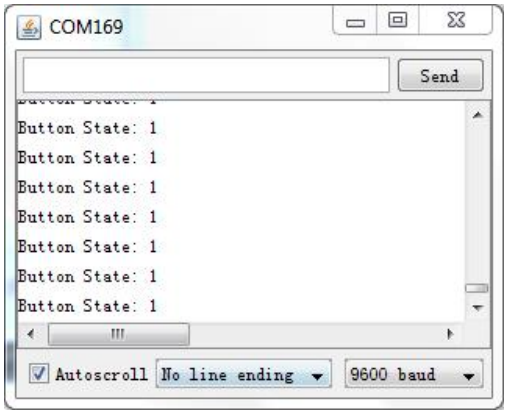
Open the Arduino Code p03_Button.ino

p03_Button §

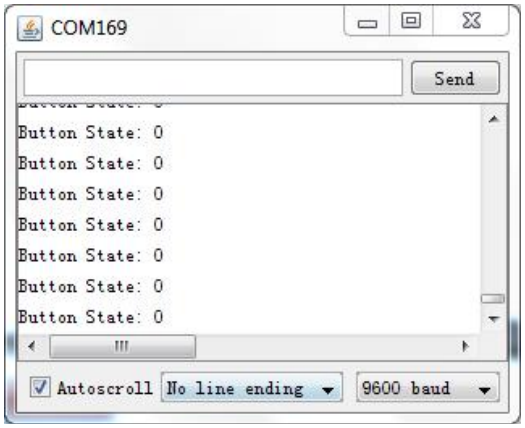
```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 3 - Button

  Detect the state of a button and display the information on Serial Monitor

  http://aliexpress.com/store/226959
*/
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;     // the pushbutton pin
int buttonState = LOW;
void setup() {
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  Serial.print("Button State: ");
  Serial.println(buttonState);
}
```

When uploading success, Click Serial Monitor tool

When the button is kept pressed, the Monitor displays



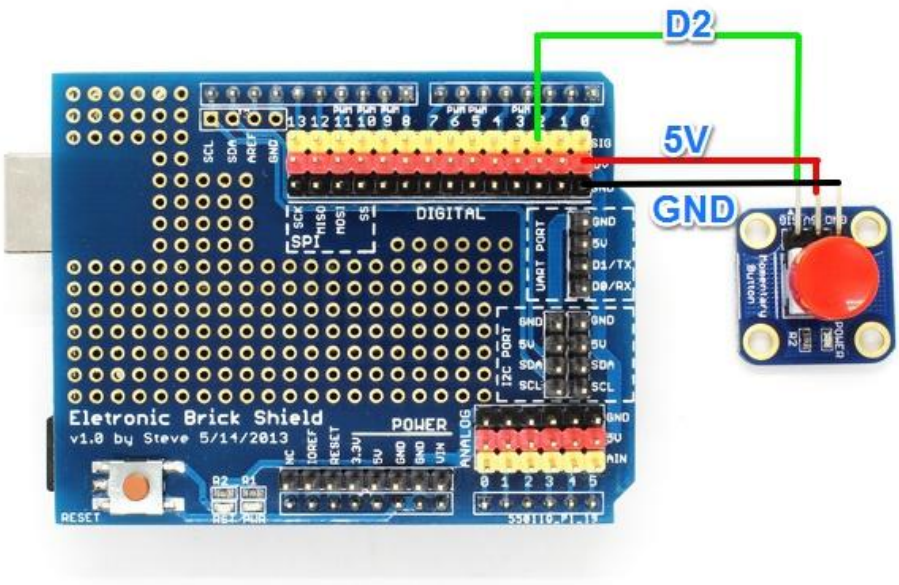
Once the button is released, the Monitor displays



Experiment – 04 Read the Momentary Button Input

Using the same Arduino code and compare the output information with the experiment – 03.

Hardware Connection

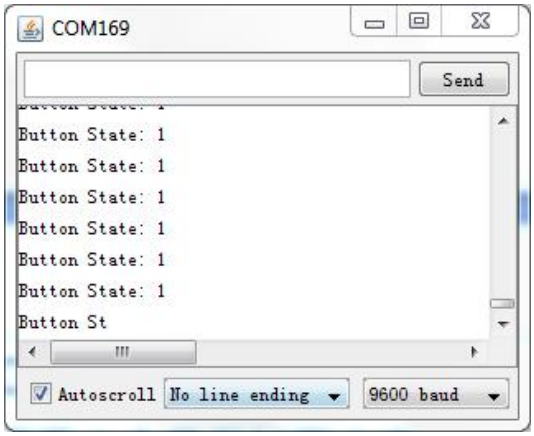


Open the Arduino Code p04_MomentaryButton.ino

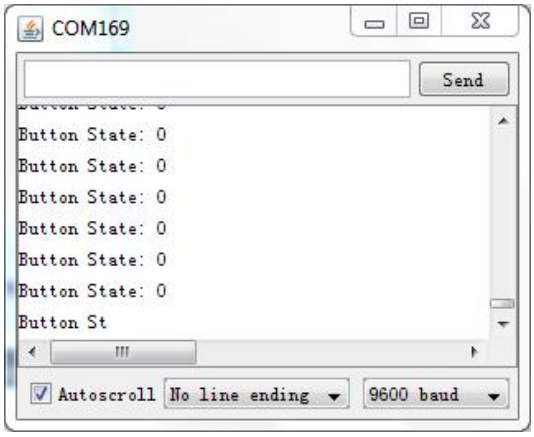
p04_MomentaryButton §

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 4 - MomentaryButton

  Detect the state of a button and display the information on Serial Monitor
  The momentary button will keep the state untill you push it again.

  http://aliexpress.com/store/226959
*/
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status
void setup() {
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  Serial.print("Button State: ");
  Serial.println(buttonState);
}
```

When successfully uploading the code, open the Serial Terminal, press the button and release your hand. The terminal displays



Press the button again, the terminal displays

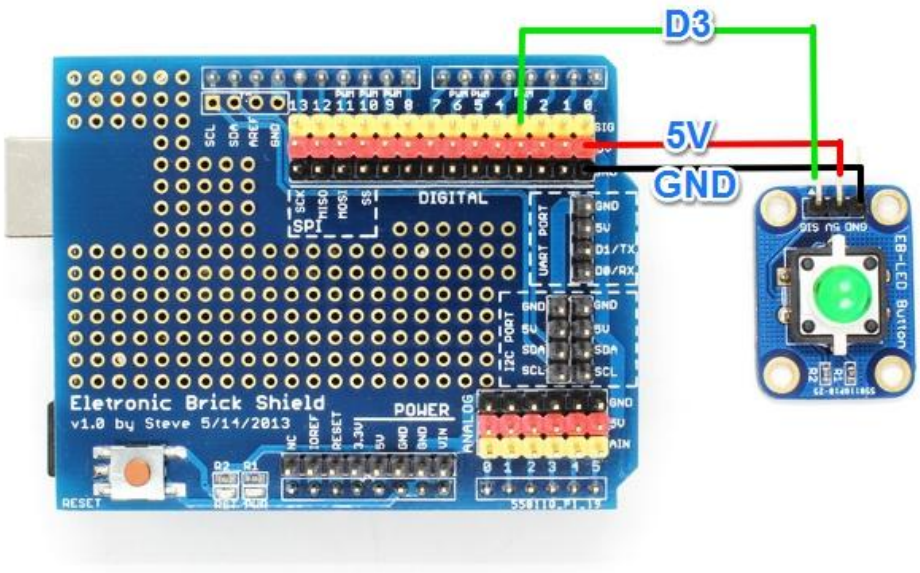


Experiment – 05 Read LED Button Input by External Interrupt

Learn to use the function of attachInterrupt(1, pushButton, RISING),

Refer here http://arduino.cc/en/Reference/AttachInterrupt

Hardware Connection

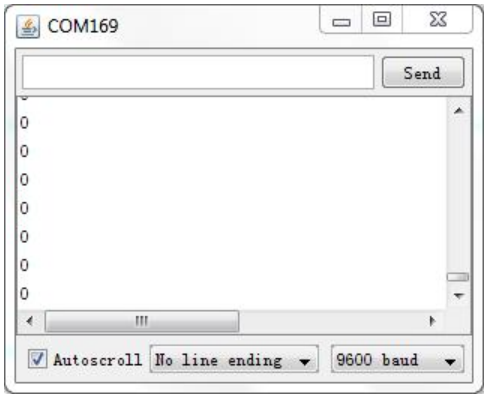


Open the Arduino Code p05_LED_Button.ino

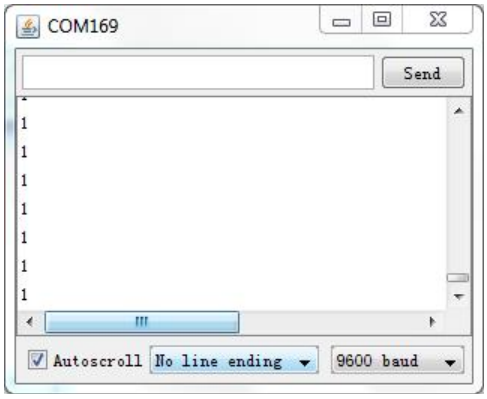
p05_LED_Button

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 5 - LED Button

  Detect the state of a LED Button and output the information on Serial Monitor

  http://aliexpress.com/store/226959
*/
volatile int state = LOW;

void setup()
{
  Serial.begin(9600);
  attachInterrupt(1, pushButton, RISING);//once button been pushed, function pushButton()
}
void loop()
{
  Serial.println(state);
}
void pushButton()
{
  state = !state;//global veriable changes it's state once button pushed
}
```

When successfully uploading the code, open the Serial Terminal, and it displays



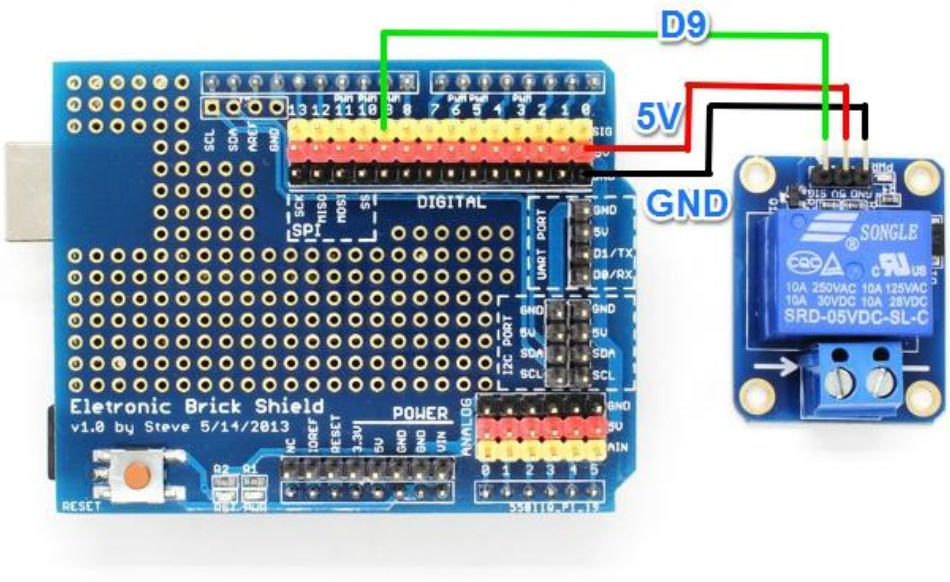
The output changes every time the button pressed.



Experiment – 06 Control the Relay

Using Relay to control High Voltage Device (like household appliance) by small signal (like Arduino) is the most convenient way.

Hardware Connection



Rating of the Relay is 250VAC/10A, 30VDC/10A. The Relay works like an electronic Switch.

CAUTION: Take care of the High Voltage on the pin of the module. You must operate under the guidance of person whom knows about how to avoid the danger. We are not responsible for any hurt or damage.

Open the Arduino Code p06_Relay.ino

p06_Relay

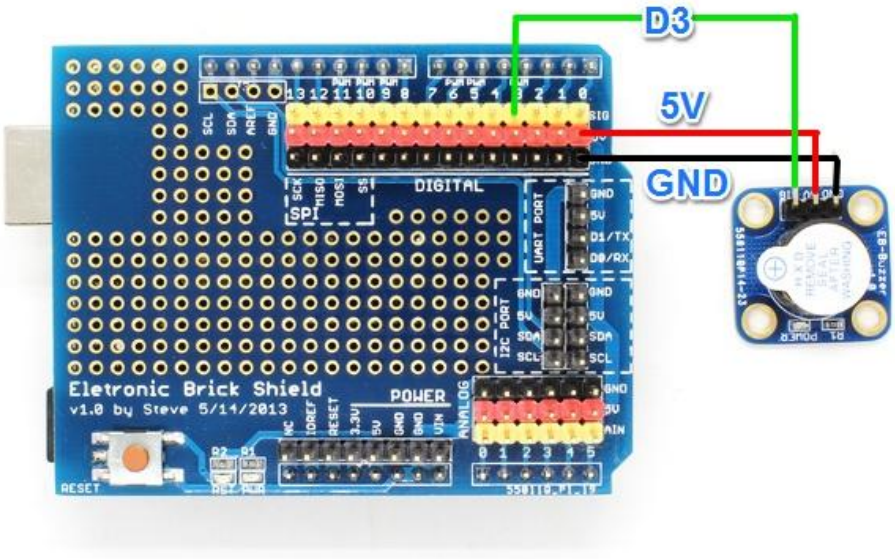
```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 6 - Relay

  Turns on an Relay for one second, then off for one second, repeatedly.

  http://aliexpress.com/store/226959
 */
// connect Relay to arduino pin 9
const int relay = 9;
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(relay, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(relay, HIGH);   // turn the Relay on (HIGH is the voltage level)
  delay(1000);                 // wait for a second
  digitalWrite(relay, LOW);    // turn the Relay off by making the voltage LOW
  delay(1000);                 // wait for a second
}
```

After successfully uploading the code, you can hear the sound from the working Relay. It is opening and closing the switch in one second interval.

Experiment – 07 Control the Buzzer

Buzzer is wided used for warning and instructions. The experiment will use Arduin IO to control the buzzer on/off, and the volume of the sound through changing the duty ratio of the PWM signal.

Hardware Connection



Open the Arduino Code p07_Buzzer.ino

p07_Buzzer

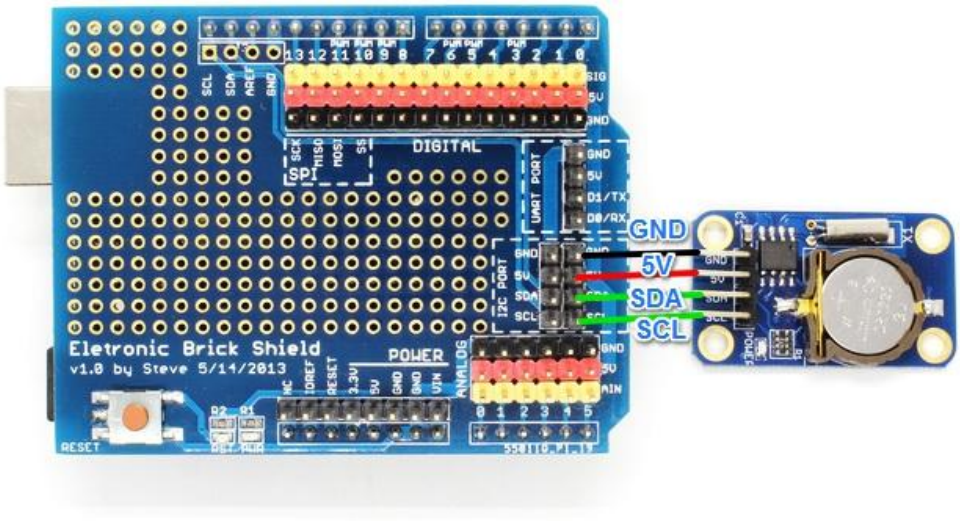
```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 7 - Buzzer

  Make some noise by buzzer

  http://aliexpress.com/store/226959
*/
const int buzzerPin = 3;
int volume = 0; // volume of the buzzering

void setup()
{
  pinMode(buzzerPin, OUTPUT); // set as OUTPUT
}

void loop()
{
  // make some noise by buzzer
  for(volume = 0; volume < 255; volume++){
    analogWrite(buzzerPin, volume);
    delay(20);
  }

  // make some noise by buzzer
  for(volume = 255; volume > 0; volume--){
    analogWrite(buzzerPin, volume);
    delay(20);
  }

  // make some noise by buzzer
  volume = 255 - volume / 2 + volume % 2;
  analogWrite(buzzerPin, volume);
  delay(1);
}
```

After successfully uploading the code, we can hear a series of changing sound.

Experiment – 08 Get Real Time Clock

The experiment uses the Brick with DS1307, which outputs the real time clock signal through standard I2C interface.

Hardware Connection



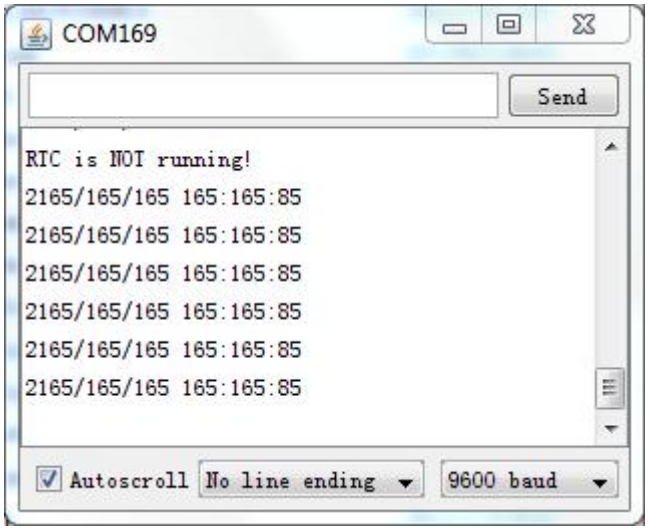
Open the Arduino Code p08_RTC.ino

p08_RTC §

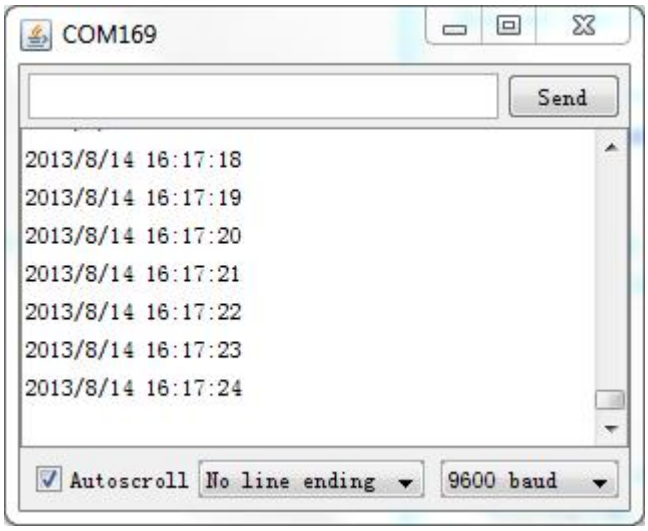
```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 8 - RTC

  Date and time functions using a DS1307 RTC connected via I2C and Wire lib

  http://aliexpress.com/store/226959
*/
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 RTC;

void setup () {
    Serial.begin(9600);
    Wire.begin();
    RTC.begin();

  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
}
```

```
void loop () {
    DateTime now = RTC.now();

    Serial.print(now.year(), DEC);
    Serial.print('/');
    Serial.print(now.month(), DEC);
    Serial.print('/');
    Serial.print(now.day(), DEC);
    Serial.print(' ');
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();

    delay(1000);
}
```

When successfully uploading the code, open the Serial Terminal, and it may display the similar information as following (The battery is uninstalled)



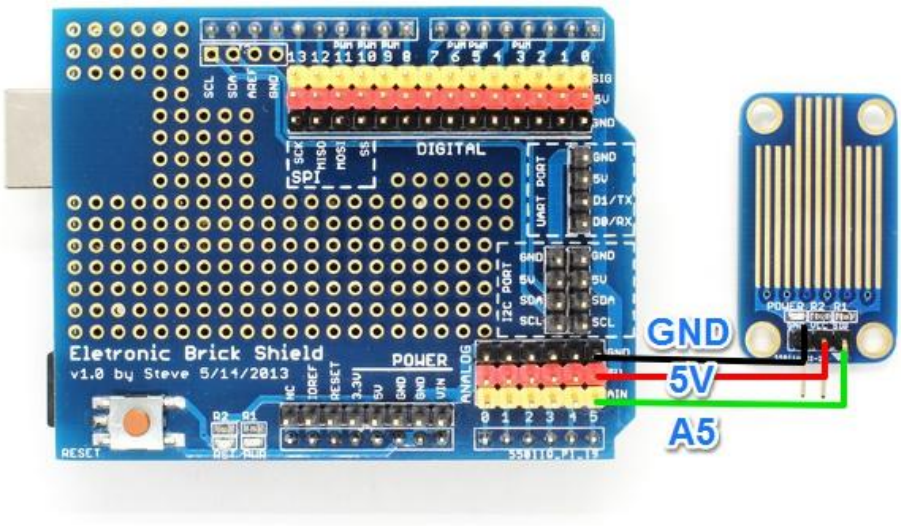
When the button battery is installed, it may display the right date and time.



Experiment – 09 Read the Water Sensor

Use water sensor to detect if there′ s water on the sensor by reading Arduino Analog pin.

Hardware Connection

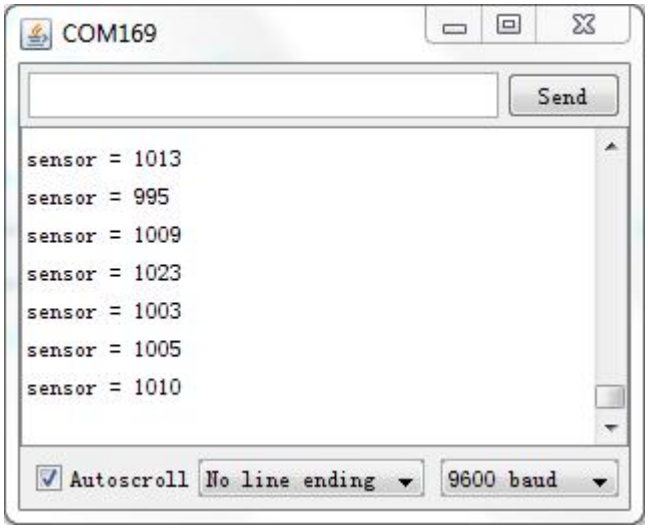


Open the Arduino Code p09_WaterSensor.ino

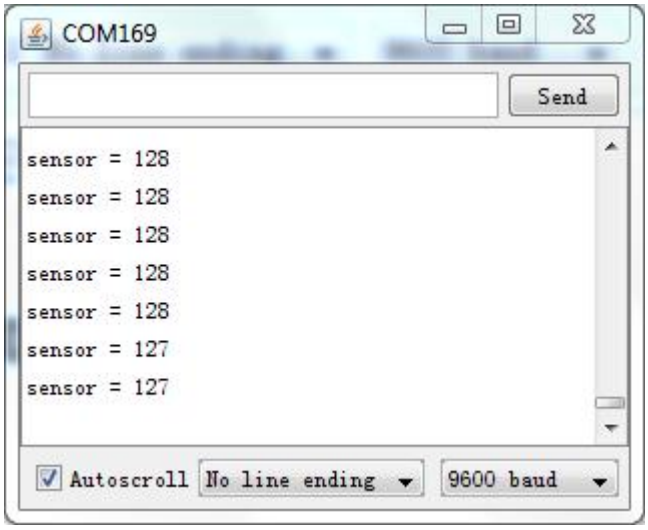
p09_WaterSensor

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 9 - Water Sensor

  See the changes when there's water and no water on the sensor.

  http://aliexpress.com/store/226959
 */
// Connect the sensor to Analog input 5
const int sensorPin = A5;
int sensorValue = 0;
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
// the loop routine runs over and over again forever:
void loop() {
  // read the analog in value:
  sensorValue = analogRead(sensorPin);
  // print the results to the serial monitor
  Serial.print("sensor = " );//see the difference when there's water or not
  Serial.println(sensorValue);
  delay(1000);// Do sampling every other second
}
```

After successfully uploading the code, when there′ s no water the Serial Terminal displays the similar information as following.



When there′ s water on the Sensor, it displays

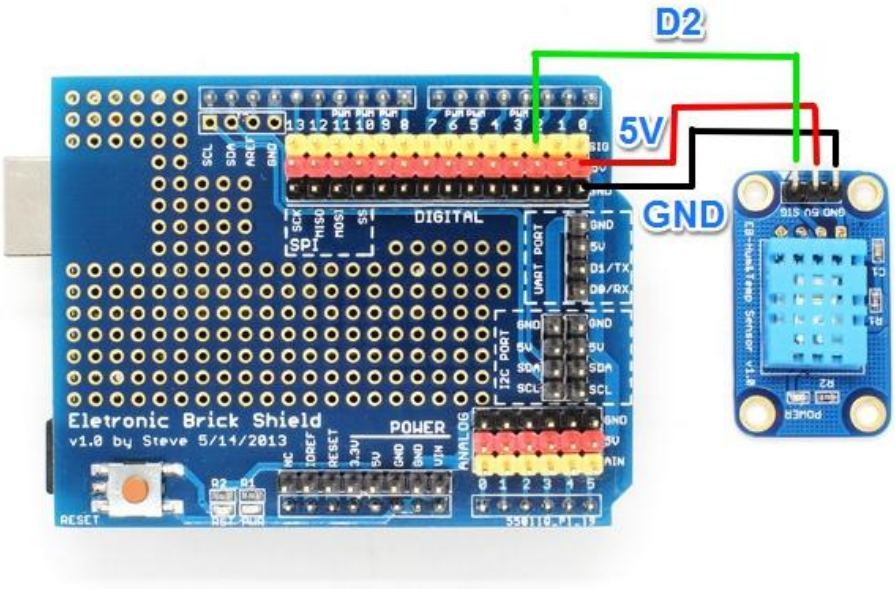


From above we can tell the difference if there′ s water or not.

Experiment -10 Read the Humidity and Temperature Sensor

Use Arduino and the sensor DHT11 to get the humidity and temperature of the environment.

Hardware Connection



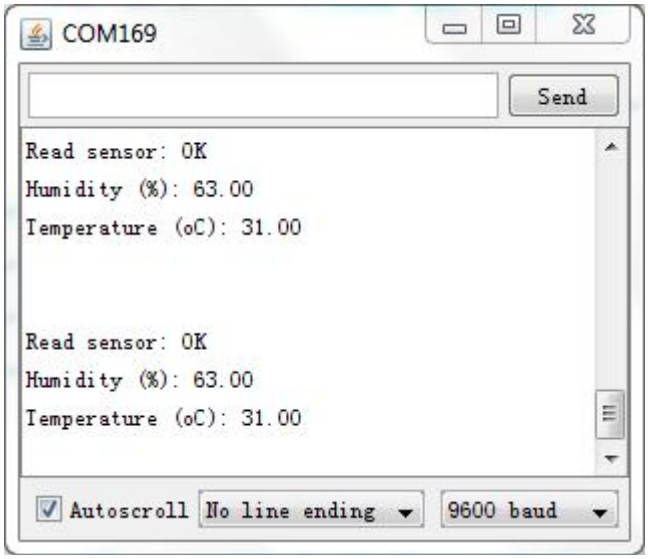
Open the Arduino Code p10_HumdAndTempSensor.ino

p10_HumdAndTempSensor

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 10 - HumAndTempSensor

  Get the humidity and temperature value from DHT11,
  and display it on the Serial Monitor.

  http://aliexpress.com/store/226959
*/
#include <Arduino.h>
#include <Wire.h>
#include <DHT11.h>

dht11 DHT11;

#define DHT11PIN 2

void setup()
{
  Serial.begin(9600);
}
```

```
void loop()
{
  Serial.print("\n");
  int chk = DHT.read(DHT11_PIN);
  Serial.print("Read sensor: ");
  switch (chk)
  {
    case DHTLIB_OK:
                Serial.println("OK"); break;
    case DHTLIB_ERROR_CHECKSUM:
                Serial.println("Checksum error"); break;
    case DHTLIB_ERROR_TIMEOUT:
                Serial.println("Time out error"); break;
    default:
                Serial.println("Unknown error"); break;
  }
  Serial.print("Humidity (%): ");
  Serial.println((float)DHT.humidity, 2);
  Serial.print("Temperature (oC): ");
  Serial.println((float)DHT.temperature, 2);
  delay(2000);
}
```

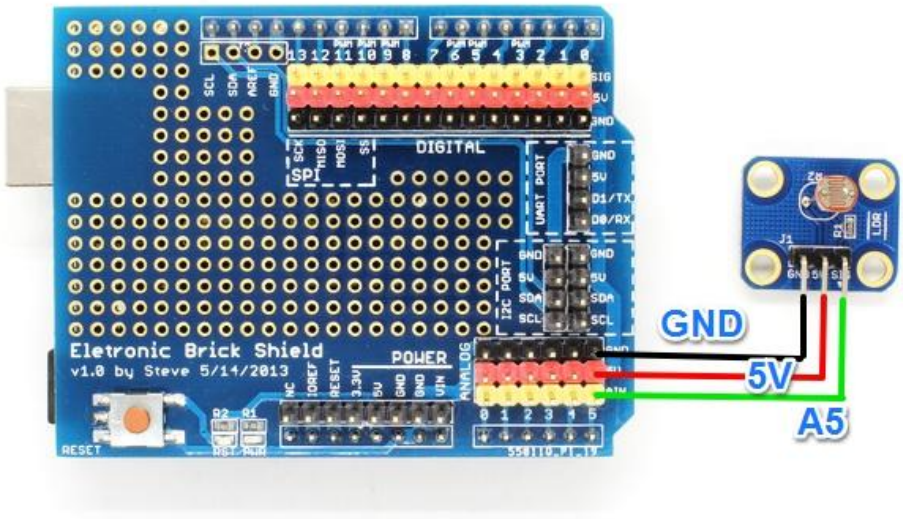
After successfully uploading the code, it outputs the following information in the Serial Terminal.



Experiment – 11 Read the Light Sensor

We can get the light information through Light Dependent Resistor (LDR)

Hardware Connection

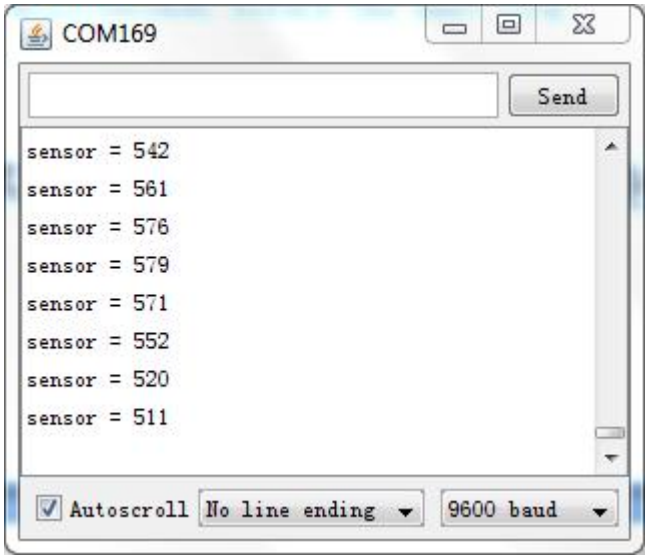


Open the Arduino Code p11_LightSensor.ino

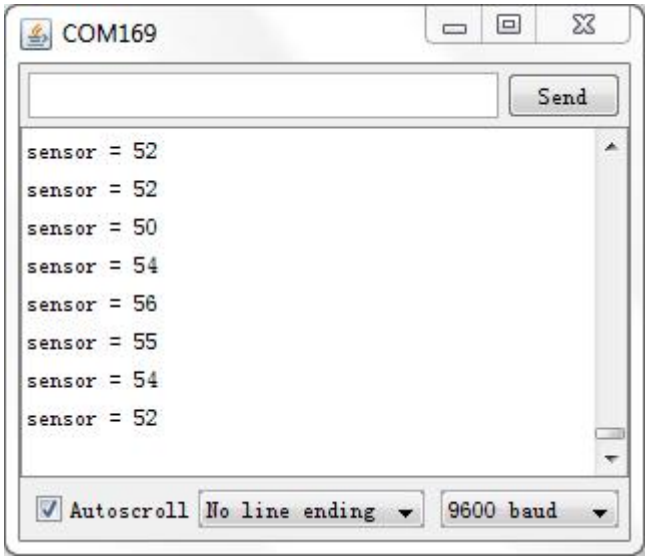
p11_LightSensor

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 11 - LightSensor

  Analog input, serial output
  Reads an analog input pin, and prints the results to the serial monitor.

  http://aliexpress.com/store/226959
 */
const int analogInPin = A5;  // Analog input pin
int sensorValue = 0;        // value read from the sensor
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.println(sensorValue);
  // wait 200 milliseconds before the next loop
  delay(200);
}
```

After successfully uploading the code, the Serial Terminal outputs the following information under the indoor environment.



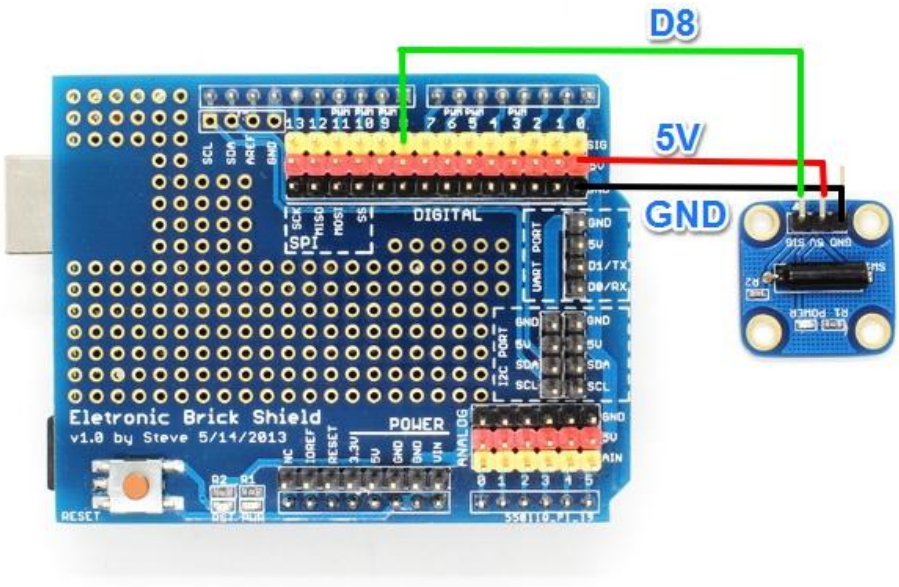
When hiding the senor by hand, the output value decreases, from which we can tell there′s little light sensed.



Experiment – 12 Read the Tilt Sensor

The experiment detect if the object is tilted or not.

Hardware Connection

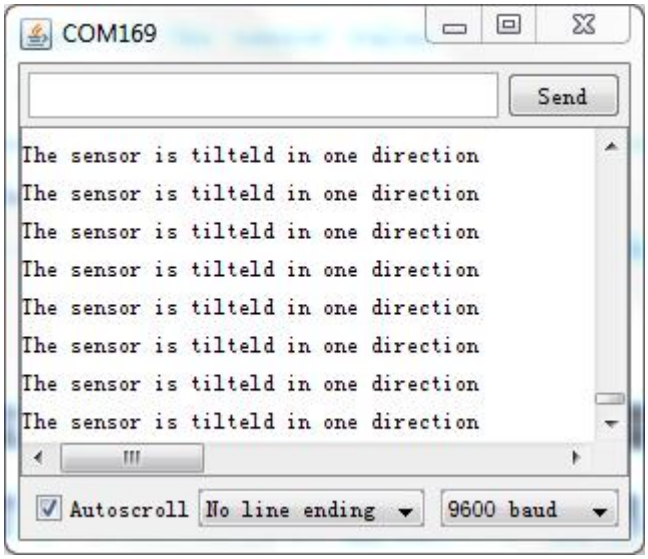


Open the Arduino Code p12_TiltSensor.ino

p12_TiltSensor

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 12 - TiltSensor

 Output the information to Serial Monitor when the sensor tilted.

 http://aliexpress.com/store/226959
 */
const int tiltSensorPin = 8;     // tilt sensor connected to pin 8
// variables will change:
int tiltState = 0;         // variable for reading the sensor status
void setup() {
  // initialize the sensor pin as an input:
  pinMode(tiltSensorPin, INPUT);
  Serial.begin(9600);
}
void loop(){
  // read the state of the sensor value:
  tiltState = digitalRead(tiltSensorPin);
  // if the sensor is tilted in one direction, it outputs HIGH:
  if (tiltState == HIGH) {
    Serial.println("The sensor is tilteld in one direction");
  }
}
```

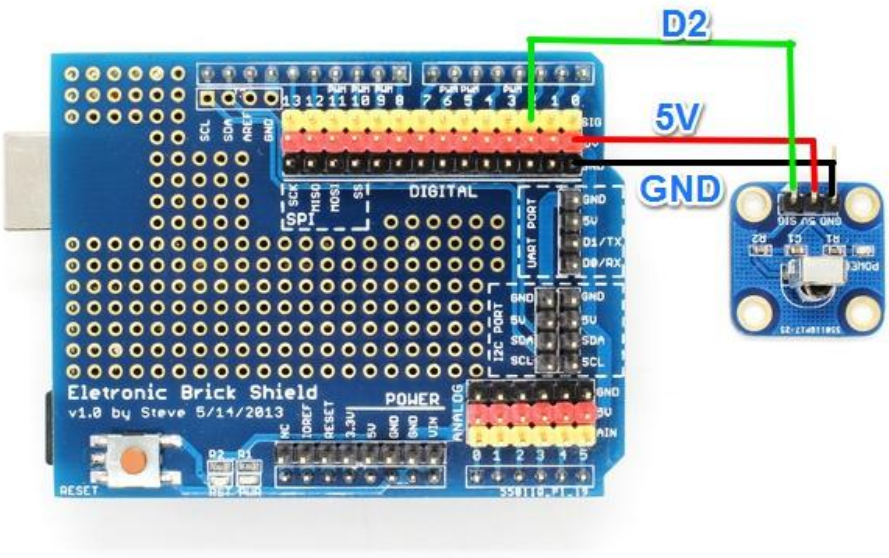
After successfully uploading the code, open the Serial Terminal, it outputs the following information when the sensor is tilted in one direction. You may note that the sensor can only detect on direction, so if you need to detect more directions, then more tilt sensors are need.


COM169
Send
The sensor is tilteld in one direction
The sensor is tilteld in one direction
The sensor is tilteld in one direction
The sensor is tilteld in one direction
The sensor is tilteld in one direction
The sensor is tilteld in one direction
The sensor is tilteld in one direction
The sensor is tilteld in one direction
Autoscroll
No line ending
9600 baud

Experiment – 13 Infrared Remote Control

Use Arduino and Infrared Sensor to receive the infrared control signal sent from the remote controller.

Hardware Connection

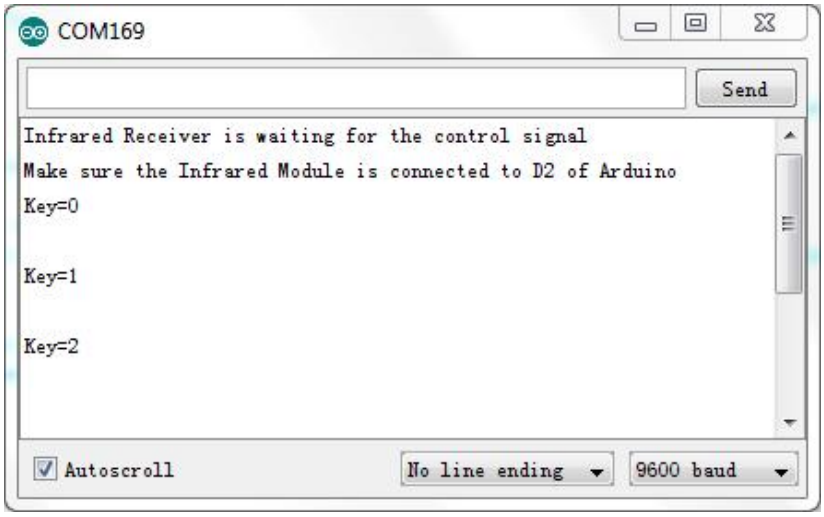


Open the Arduino Code p13_IR_RemoteControl.ino

```
p13_IR_RemoteControl

/*
 Arduino Electronic Brick Advanced Kit example
 Project 13 - IR Remote Control

 Receive the signal from the IR controller, and output the key information

 http://www.xxxxxxxx.com/xxxx/xxxxx.html
 */
#include <xxxx.h>
#include <xxx.h>
#include "IRremote.h"

int key=0;//present key
//The code must interrupt 0, and it works only when the module is connected to D2 of Arduino
IRrecv irrecv(2);
void setup() {
  irrecv.enableIRIn();
  Serial.begin(9600);
  Serial.println("IRremote Receiver is waiting for the control signal");
  Serial.println("Make sure the IRremote Module is connected to D2 of Arduino");
}

void loop() {
  key = irrecv.getCode();// get the present key information
  if (key != 0)
  {
    Serial.print("Key=");//Display the key information
    Serial.println(key);
    Serial.println();
  }
}
```

After successfully uploading the code, open the Serial Terminal, it will display the key information when the key pressed on the remote controller. (Note: you need to take out the plastic pad to allow the battery work)

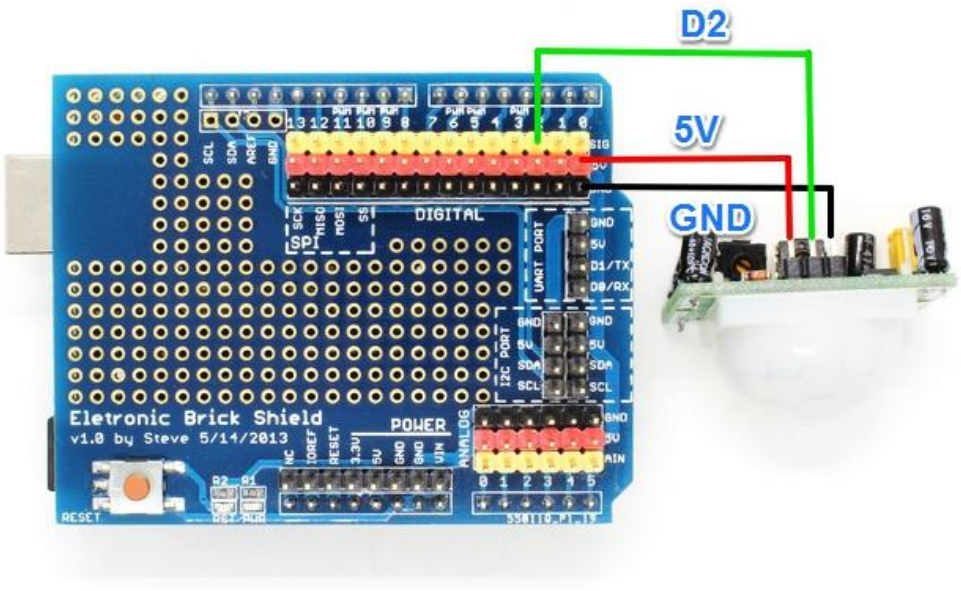

CH-
CH
CH+
EQ
0
100+
200+
1
2
3
5
6
VELLEMAN


COM169
Send
Infrared Receiver is waiting for the control signal
Make sure the Infrared Module is connected to D2 of Arduino
Key=0
Key=1
Key=2
Autoscroll
No line ending
9600 baud

Experiment – 14 PIR motion Sensor

Arduino can detect a moving human body by PIR motion sensor. The sensor outputs high level, and Keep for a while, and then go low level.

Hardware Connection



Open the Arduino code p14_PIRMotionSensor.ino

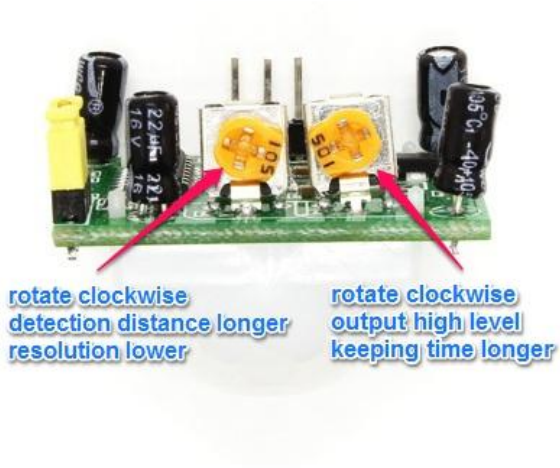
p14_PIRMotionSensor

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 14 - PIRMotionSensor

  Output the state when detecting people move nearby.

  http://aliexpress.com/store/226959
 */
const int PIRSensorPin = 2;     // the number of the Sensor pin

// variables will change:
int sensorState = 0;         // variable for reading the Sensor status

int flag = 0; //use the flag to prevent continuous outputing when no moving people

void setup() {
  // initialize the Sensor pin as an input:
  pinMode(PIRSensorPin, INPUT);
  digitalWrite(PIRSensorPin, LOW);
  Serial.begin(9600);
}
```

```
void loop() {
  // read the state of the Sensor value:
  sensorState = digitalRead(PIRSensor);

  // check if the moving people sensor
  // if it is, the sensorState is HIGH:
  if (sensorState == HIGH) {
    // turn LED on
    Serial.println("Moving people detected");
    lastState = 0;
  }
  else {
    if (lastState == 0) {
      Serial.println("No moving people");
      lastState = 1;
    }
  }
}
```

After successfully uploading the code, open the Serial Terminal, it will display the following information. When there is not moving people, it outputs "No moving people". When there is moving people, it keeps outputting "Moving people detected".

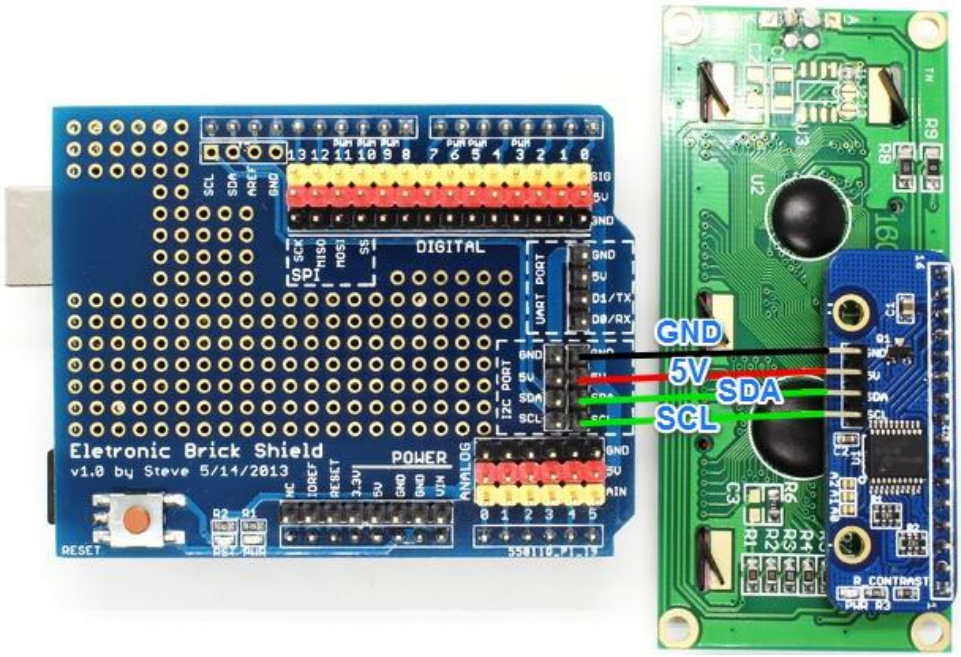
The detection distance and the output high level keeping time can be adjusted by rotating the potentiometer.



Experiment – 15 Display Characters on I2C 1602 LCD

Use Arduino control I2C 1602 LCD and display characters. The hardware is updated based on the product from Adafruit and the library is open source.

Hardware Connection



Open the Arduino Code p15_I2C_1602LCD.ino

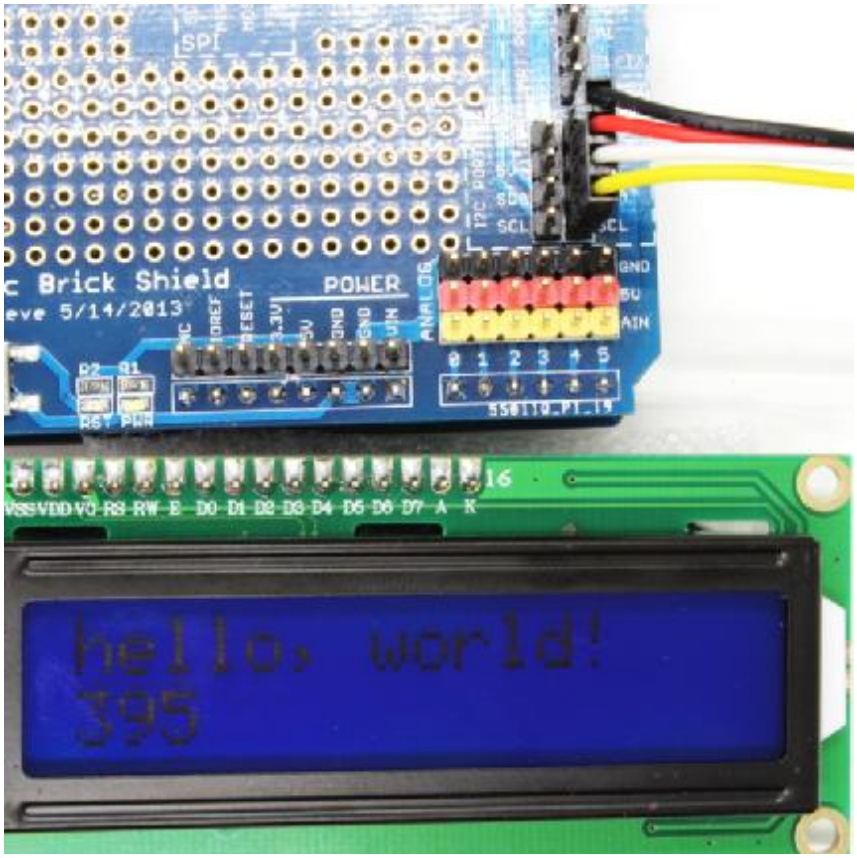
p15_I2C_1602LCD

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 15 - I2C 1602 LCD

 This sketch prints "Hello World!" to the LCD
 and shows the time.

  The circuit:
 * 5V to Arduino 5V pin
 * GND to Arduino GND pin
 * CLK to Analog #5
 * DAT to Analog #4

  http://aliexpress.com/store/226959
*/

// include the library code:
#include <Wire.h>
#include "LiquidCrystal.h"
```

```
// Connect via i2c, default address #0 (A0-A2 not jumpered)
LiquidCrystal lcd(0);

void setup() {
  // set up the LCD's number of rows and columns:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);

  lcd.setBacklight(HIGH);
  delay(500);
  lcd.setBacklight(LOW);
  delay(500);
}
```

After successfully uploading the code, the LCD displays characters when the backlight is on.

The LCD displays characters when the backlight is off.

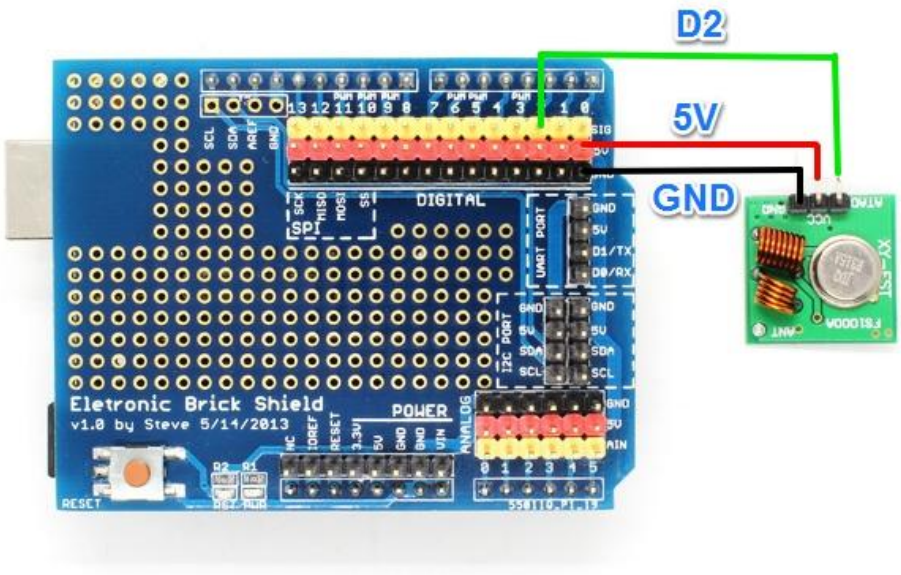

SPI
Brick Shield
eve 5/14/2013
POWER
IOREF
RESET
3.3V
5V
GND
GND
VIN
ANALOG
SCL
GND
5V
AIN
0 1 2 3 4 5
D2
D1
RST
PWR
VSS VDD VO RS RW E D0 D1 D2 D3 D4 D5 D6 D7 A K
16
hello, world!
395

Experiment -16 RF Transmitter

Note: To successfully implement the following two experiment of RF communication, you need two Arduino board and EB Shield.

Use Arduino and RF module to transmitter some message to the receiver.

Hardware Connection



Open Arduino Code p16_315M_RF_Transmitter.ino and uploading

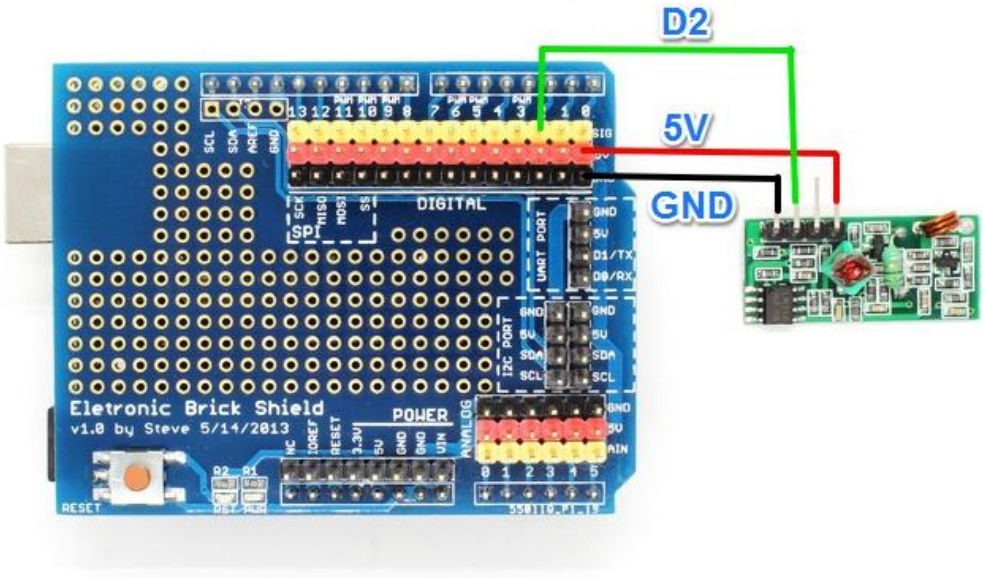
p16_315M_RF_Transmitter

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 16 - 315MHz RF Transmitter

  Tansmit message "Hello" to the Receiver every other second

  http://aliexpress.com/store/226959
 */
#include <VirtualWire.h>

int RF_TX_PIN = 2;

void setup()
{
  vw_set_tx_pin(RF_TX_PIN); // Setup transmit pin
  vw_setup(2000); // Transmission speed in bits per second.
}
void loop()
{
  const char *msg = "Hello";
  vw_send((uint8_t *)msg, strlen(msg)); // Send "Hello" every 1000ms.
  delay(1000);
}
```

Experiment – 17 RF Receiver

Use Arduino and RF module to receive the message sent from the transmitter and outputs to the Serial Terminal.

Hardware Connection



Open the Arduino Code p17_315M_RF_Receiver.ino

p17_315M_RF_Receiver

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 17 - 315MHz RF Receiver

  Wait to receive message sent from the transimtter

  http://aliexpress.com/store/226959
 */

#include <VirtualWire.h>

int RF_RX_PIN = 2;

void setup()
{
  Serial.begin(9600);
  Serial.println("setup");
  vw_set_rx_pin(RF_RX_PIN);  // Setup receive pin.
  vw_setup(2000); // Transmission speed in bits per second.
  vw_rx_start(); // Start the PLL receiver.
}

void loop()
{
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
  if(vw_get_message(buf, &buflen)) // non-blocking
  {
    int i;
    // Message with a good checksum received, dump it.
    Serial.print("Got: ");
    for(i = 0; i < buflen; ++i)
    {
      Serial.print(char(buf[i]));
      //Serial.print(" ");
    }
    Serial.println("");
  }
}
```

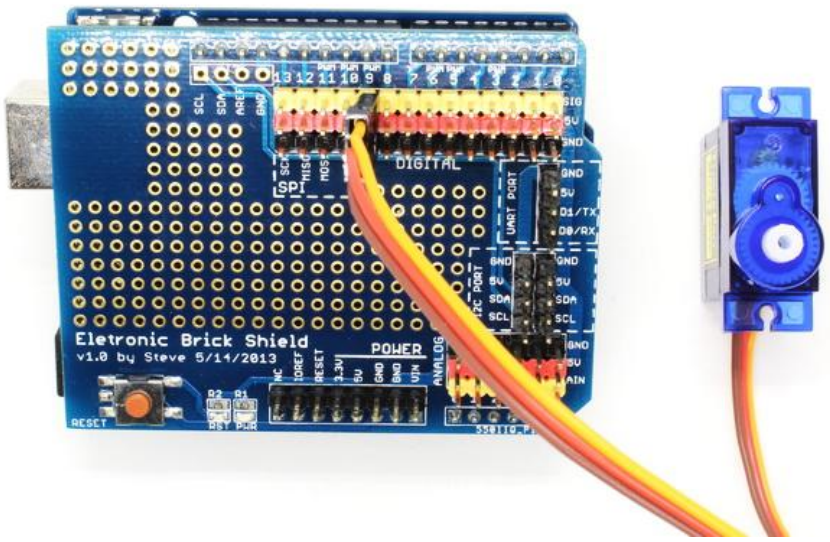
After successfully uploading the code, open the Serial Terminal, we can receive the following information.

Experiment - 18 Control Servo Motor

Use Arduino to control 9g servo motor rotate between 0~180 degree.

Hardware Connection

Connect Servo to D9



Open Arduino Code p18_Servo.ino

p18_Servo

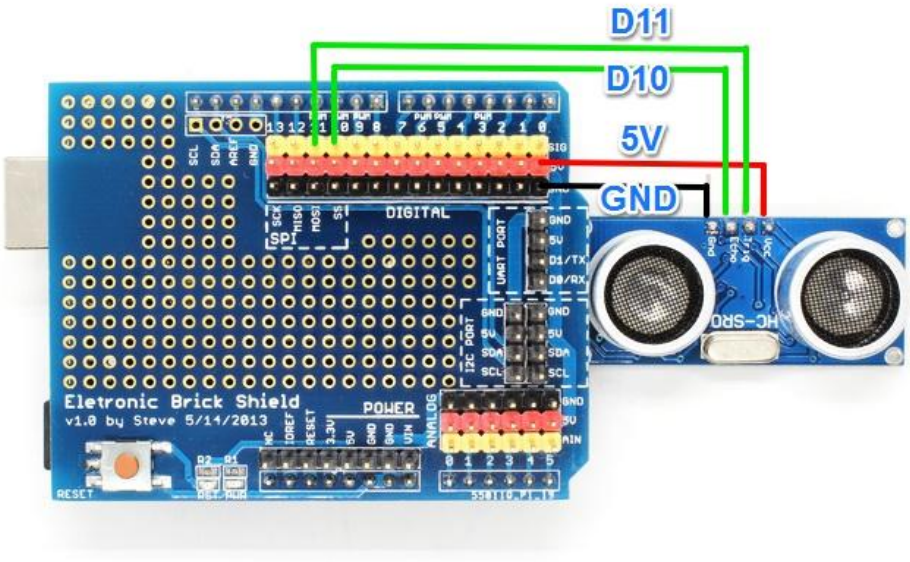
```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 18 - Servo

  Sweep

  by BARRAGAN <http://barraganstudio.com>
  This example code is in the public domain.

  http://aliexpress.com/store/226959
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
                // a maximum of eight servo objects can be created

int pos = 0;    // variable to store the servo position

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees to 180 degrees
  {                                  // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=1; pos-=1)     // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

After successfully uploading the code, the gear of the servo rotates between 0~180 degree.

Experiment – 19 Measure Distance by Ultrasonic Sensor

Use Arduino and ultrasonic sensor to measure distance.

Hardware Connection



Open the Arduino Code p19_Ultrasonic.ino

```
p19_Ultrasonic

/*
  Arduino Electronic Brick Advanced Kit example
  Project  9 - Ultrasonic

  Measure the distance by HC-SR04 Ultrasonic Ranging Module

  http://www.seeedstudio.com/depot/2280.html
*/

#include "Ultrasonic.h"

Ultrasonic ultrasonic(11,10);//(Trig,Echo)

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    //output the distance in cm
    Serial.println(ultrasonic.Ranging(CM));
    delay(200);
}
```

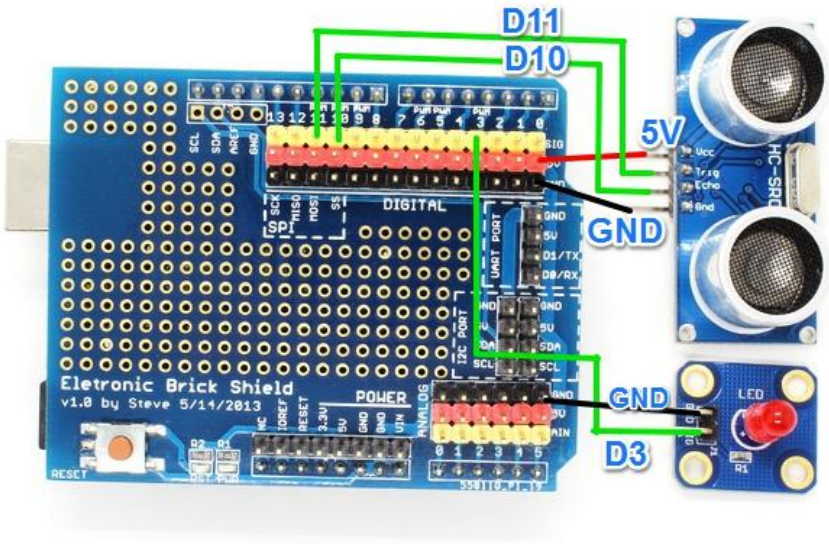
We can get the reliable result within the range of 3m.

After successfully uploading the code, open the Serial Terminal and it displays the distance between the ultrasonic sensor and the target.

Experiment – 20 Distance controlled Breathing LED Light

You can use your hand to control the brightness of the LED by changing the distance closed to the Ultrasonic Sensor. To get a good effect, the distance should be within 20cm. The closer the LED is brighter.

Hardware Connection



Open the Arduino Code p20_DistanceControledLED.ino

p20_DistanceControledLED

```
/*
  Arduino Electronic Bricks Advanced Kit example
  Project 20 - Distance Controled LED

  Change the lighting strength based on the distance of closing object

  http://aliexpress.com/store/226959
*/

#include "Ultrasonic.h"

Ultrasonic ultrasonic(11,10);//(Trig,Echo)

int distance = 0;
const int LED = 3;//Digital pin 3 with PWM function
int outputValue = 0;        // value output to the PWM (analog out)

  void setup() {
        Serial.begin(9600);
  }

  void loop()
  {
        //Read the distance value
        distance = ultrasonic.Ranging(CM);
        Serial.print(distance);

        if(distance < 20){
          outputValue = map(distance, 0, 20, 255, 0);
          // change the analog out value
          analogWrite(LED, outputValue);
        }else{
          digitalWrite(LED, LOW);
        }
        delay(100);
  }
```

Upload the code, and change the distance between the Ultrasonic Sensor and your hand you may find the variance of the bright LED.

Reference Link

Arduino functions http://arduino.cc/en/Tutorial/HomePage