# 4duino Sensor Kit 40-in-1

Dear customer,

Thank you for choosing our 4duino Sensor Kit.

The 4duino Sensor Kit has been re-installed by us and the boards have been developed especially for the most common open source platforms. The high compatibility distinguishes this Sensor Kit.

The following instructions contain the technical description for the individual sensors, the pin assignment or the respective chipset used.

The Sensor Kit does not include the connecting cables or other accessories.

The Sensor Kit provided by us is suitable for beginners and experienced hobbyists, who are interested in electronics and want to design their own ideas and circuits.

We hope you enjoy the sensors and your experiments.

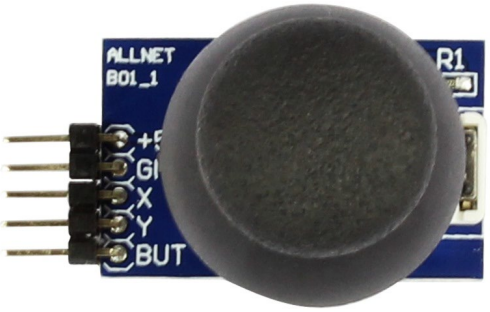
Your ALLNET Team

ALLNET 4duino

## Table of Contents Sensor Overview

# Sensor Overview

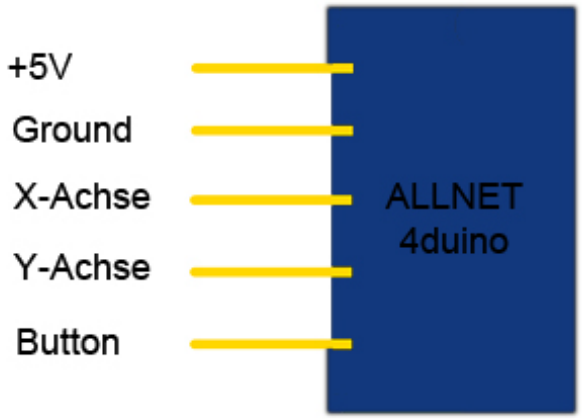
## B01 Joystick Module (XY Axes)




## Brief Description / Technical Data

In this program, the values of the joystick (X and Y axes as well as the button (Z)) are read out 10 times per second and output via the serial interface.

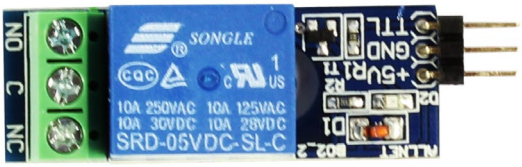
## Pin Assignment



## Code Sample

```
// ALLNET Joystick B01
// Information http://www.allnet.de

//Declaring the necessary variables
int JoyStick_X = A0;
int JoyStick_Y = A1;
int JoyStick_Button = 3;

//one-time setup commands
void setup ()
{
  //Allocate pin function
  pinMode (JoyStick_X, INPUT);
  pinMode (JoyStick_Y, INPUT);
  pinMode (JoyStick_Button, INPUT_PULLUP);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
```

# B02 5V Relay Module



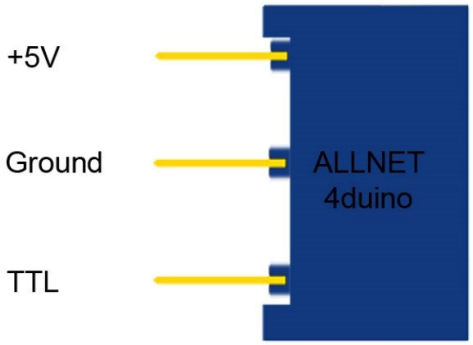

## Brief Description / Technical Data

Voltage range: 250VAC / 10A | 30VDC / 10A
Relay for switching higher voltages by means of a 5V output.
The output bar of the relay has two output terminals:

- One of them is marked with "NC" for "normally closed", which means that this path is short-circuited by default without electrical switchover at the relay.
- The other one is marked with "NO" for "normally open", which means that this path is normally open or disconnected at the relay without electrical switchover at the relay.


normally closed: Ausgangszustand geschlossen, Strom fließt
normally open: Ausgangszustand geöffnet, kein Stromfluss

## Pin Assignment




## Code Sample

```
// ALLNET 5V Relay Module B02
// Information http://www.allnet.de

//Declaring the necessary variables
int releyPin = 10;

//one-time setup commands
void setup ()
{
  //Allocate pin function
  pinMode (releyPin, OUTPUT);
}

//permanently repeated main loop
void loop ()
{
  //Applying 5V (digital HIGH) to releyPin
  digitalWrite (releyPin, HIGH);
  //Pause
  delay (3000);
  //Applying 0V (digital LOW) to releyPin
  digitalWrite (releyPin, LOW);
  //Pause
  delay (3000);
}
```

# B03 Microphone Sensor Module


## Brief Description / Technical Data

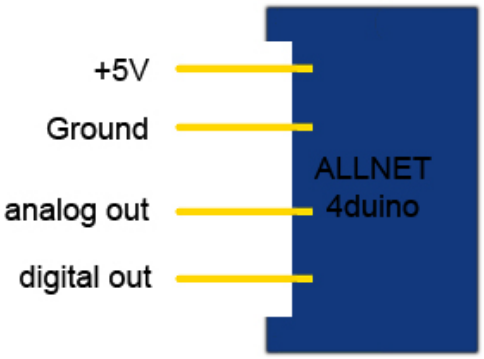
In this circuit, the microphone is read out and the readout value is output via the serial interface.

Analog output: Direct microphone signal as voltage level.

LED1: Indicates that the sensor is powered.

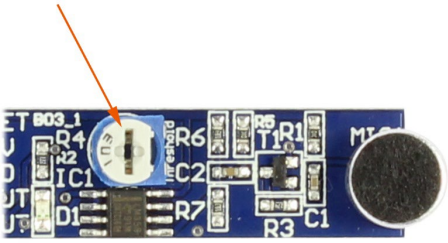
LED2: Indicates that a magnetic field has been detected.

## Pin Assignment




## How the sensor works

The sound sensor does not return fixed or relative values. It amplifies the signal depending on the set resistance at the rotary potentiometer and directs it to the analog output of the module.




## Code Sample

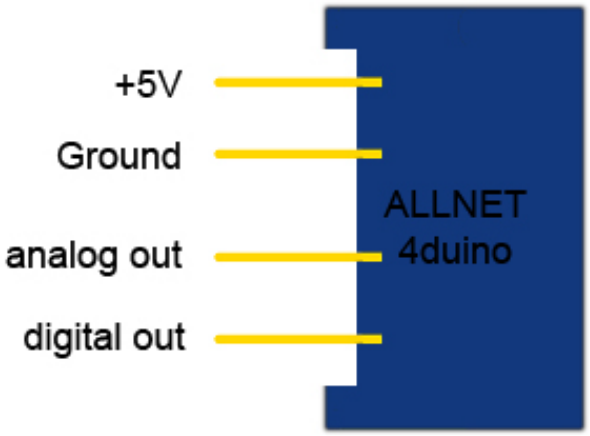
```
// ALLNET Microphone Sensor Module B03
// Information http://www.allnet.de


int sensorPin = A0; // select the input pin for the potentiometer
int sensorValue = 0; // variable to store the value coming from the sensor

void setup () {
  pinMode (ledPin, OUTPUT);
  Serial.begin (9600);
}

void loop () {
  sensorValue = analogRead (sensorPin);
  Serial.println (sensorValue, DEC);
}
```

# B04 IR Optical Detection


## Brief Description / Technical Data

With this circuit, the values of the IR sensor are output via the serial interface.

## Pin Assignment




## How the sensor works

The module consists of a sensor unit or receiver diode, the potentiometer for setting the sensitivity and output via the pins.

## Code Sample

The program reads out the current voltage value, which can be measured at the analog output and outputs it on the serial interface.

In addition, the status of the digital pin in the console is also indicated, which means whether or not the limit has fallen below the limit value.

```
// ALLNET IR Detection B04
// Information http://www.allnet.de

int a,b,c;
void setup() {
Serial.begin(9600);
pinMode(6,OUTPUT);
}

void loop() {
  digitalWrite(6,HIGH);
  delayMicroseconds(500);
  a=analogRead(A3);
  digitalWrite(6,LOW);
  delayMicroseconds(500);
  b=analogRead(A3);
  c=a-b;
  Serial.println(c);
}
```

## B05 Flame Sensor

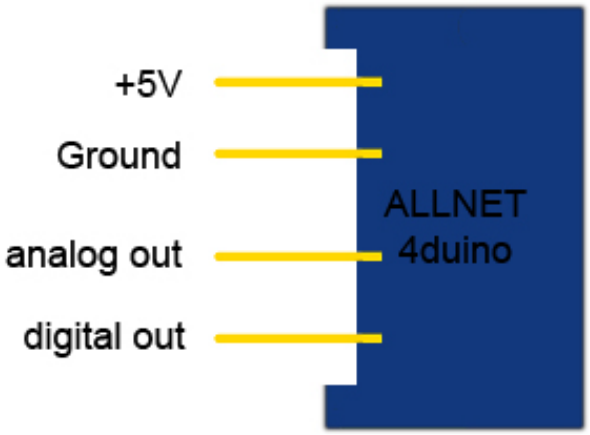

## Brief Description / Technical Data

This circuit reads the analog flame sensor and outputs the values via the serial interface. Further remarks can be found in the source code.

LED1: Indicates that the sensor is powered.

LED2: Indicates that a flame has been detected.

## Pin Assignment




## How the sensor works

The flame sensor is very sensitive to IR wavelength at 760nm - 1100nm light.

Analog output (A0): Real-time output voltage signal at the thermal resistor.

Digital output (D0): When the temperature reaches a certain threshold, the output high and low can be set via potentiometer.

## Code Sample

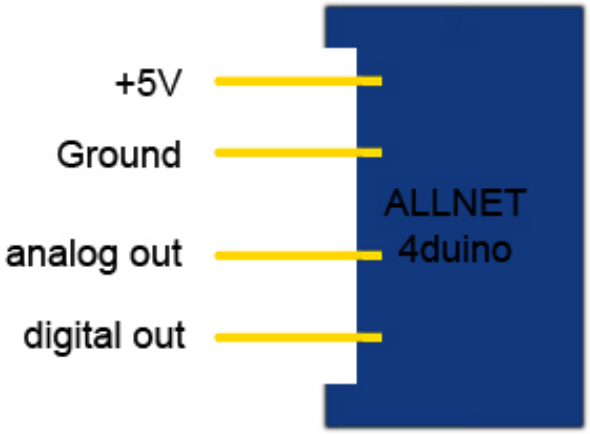
```
// ALLNET Flame Sensor B05
// Information http://www.allnet.de


const int sensorMin = 0;     // sensor minimum
const int sensorMax = 1024;  // sensor maximum

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

void loop() {
  int sensorReading = analogRead(A0);
  int range = map(sensorReading, sensorMin, sensorMax, 0, 3);

  switch (range) {
  case 0:    // A fire closer than 1.5 feet away.
    Serial.println("** Close Fire **");
    break;
  case 1:    // A fire between 1-3 feet away.
    Serial.println("** Distant Fire **");
    break;
  case 2:    // No fire detected.
    Serial.println("No Fire");
    break;
  }
  delay(1);  // delay between reads
}------------------------------");

  //Pause
  delay (200);
}
```

## B06 Hall TTL Sensor


## Brief Description / Technical Data

A hall sensor reacts to magnetic fields and converts them into electrical impulses.
Hall sensors are used for non-contact measurement of magnetic fields.

## Pin Assignment




## How the sensor works

The magnetic hall sensor reacts to a magnetic field and, depending on how it is polarized (+/-), the sensor reacts.

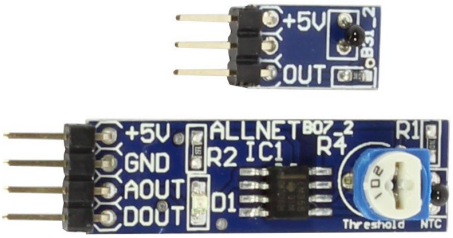
## Code Sample

The program reads out the current voltage value, which can be measured at the analog output and outputs it on the serial interface.

In addition, the status of the digital pin in the console is also indicated, which means whether or not the limit has fallen below the limit value.

```
// ALLNET Hall TTL Sensor B06
// Information http://www.allnet.de


int Led = 2 ; // define LED Interface
int buttonpin = 3; // define the linear hall magnetic sensor interface
int val ; // define numeric variables val
void setup () {
  pinMode (Led, OUTPUT) ; // define LED as output interface
  pinMode (buttonpin, INPUT) ; // define linear hall magnetic sensor output interface
}
void loop ()
{
  val = digitalRead (buttonpin) ; // digital interface will be assigned a value of 3 to
read val
  if (val == HIGH) { // When the linear hall sensor detects a magnetic signal, LED
flashes
    digitalWrite (Led, HIGH);
    Serial.println("Magnetic signal detected");
  } else {
    digitalWrite (Led, LOW);
    Serial.println("No magnetic signal detected");
  }
}
```

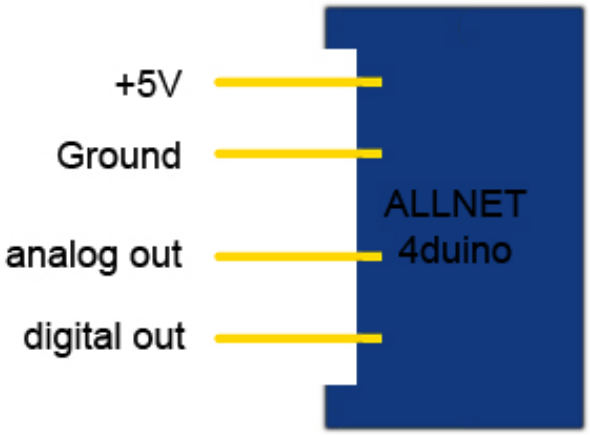
## B07/B31 NTC Threshold TTL/ NTC 10k




## Brief Description / Technical Data

The sensor is a thermal resistor that can be used to determine the temperature. The analog measured values are output to the computer via the serial interface.

## Pin Assignment




## How the sensor works

An NTC thermistor is a temperature-dependent resistor which, unlike a DHT11 temperature sensor, can be used at higher temperatures.

Temperature range:

| | |
|---|---|
| **Min/Max** | -55° to 150°C |
| **Tolerance** | approx. ±0.4°C |

## Code Sample

```
// ALLNET NTC Threshold TTL/NTC 10k B07/B31
// Information http://www.allnet.de

#define ABSZERO 273.15
#define MAXANALOGREAD 1023.0
#define ANALOGPIN A0

float temperature_NTCB(float T0, float R0, float B,
float RV, float VA_VB) {
  T0+=ABSZERO;  // convert Celsius in absolute
temperature
  float RN=RV*VA_VB / (1-VA_VB); // current NTC
resistance
  return T0 * B / (B + T0 * log(RN / R0))-ABSZERO;
}

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

void loop() {
  float T0=25;    // Nominal temperature NTC resistor
in °C
  float R0=10000; // Nominal resistance NTC sensor in
ohms
  float B=3976;   // Material constant B
  float RV=10000; // pre-resistance in ohms
  float temp;
  int aValue=analogRead(ANALOGPIN);
  // calculate with known material constant B;
  temp=temperature_NTCB(T0, R0, B, RV,
aValue/MAXANALOGREAD);
  Serial.print("NTCB:
");Serial.print(temp);Serial.println(" C");
  delay(500);
}
```

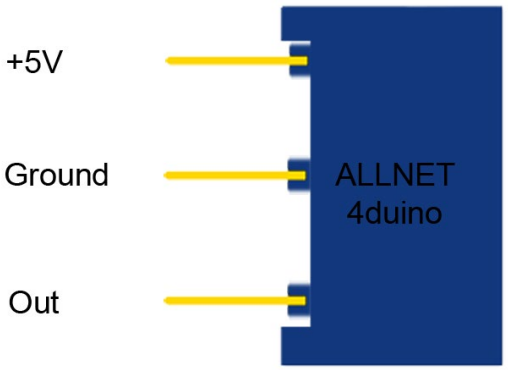
# B08 Touch Sensor


## Brief Description / Technical Data

The sensor emits a signal at the OUT pin when touched on the two silver contacts.

The program passes touching the sensor to the serial monitor.

## Pin Assignment



ALLNET 4duino

# Code Sample

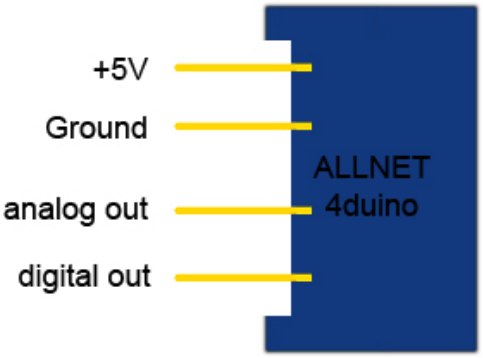
```
// ALLNET Touch Sensor B08
// Information http://www.allnet.de


//Declaring the necessary variables
int Digital_Input = 3;

//one-time setup commands
void setup ()
{
  //Allocate pin function
  pinMode (Digital_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //When the value of Digital_Input equals 1
if (digitalRead (Digital_Input) == 1)
  {
    //Then touch is detected and this is output as
message
    Serial.println ("Touch detected");
  }

  //Pause
  delay (200);
}
```

# B09 RGB LED


## Brief Description / Technical Data

LED module containing a red, blue and green LED. These are interconnected by a common cathode.

## Pin Assignment

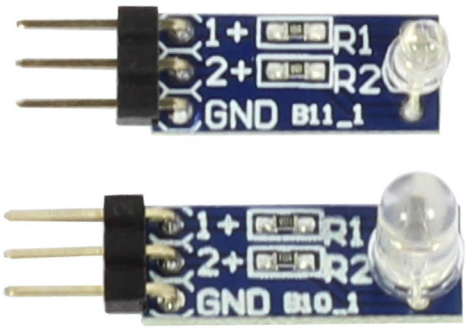


# Code Sample

This code sample shows how the integrated LEDs can be switched on alternately, in 3 second intervals, by means of a definable output pin.

```
// ALLNET RGB LED B09
// Information http://www.allnet.de


int led_blue = 5;
int led_green = 6;
int led_red = 7;

void setup() {
  Serial.begin(9600);
  pinMode(led_blue, OUTPUT);
  pinMode(led_green, OUTPUT);
  pinMode(led_red, OUTPUT);
}

void loop() {
  digitalWrite(led_blue, HIGH);
  delay(500);
  digitalWrite(led_blue, LOW);
  digitalWrite(led_green, HIGH);
  delay(500);
  digitalWrite(led_green, LOW);
  digitalWrite(led_red, HIGH);
  delay(500);
  digitalWrite(led_red, LOW);
  digitalWrite(led_blue, HIGH);
  digitalWrite(led_green, HIGH);
  delay(500);
  digitalWrite(led_blue, LOW);
  digitalWrite(led_red, HIGH);
  delay(500);
  digitalWrite(led_green, LOW);
  digitalWrite(led_blue, HIGH);
  delay(500);
  digitalWrite(led_red, LOW);
  digitalWrite(led_blue, LOW);
  delay(500);
}
```

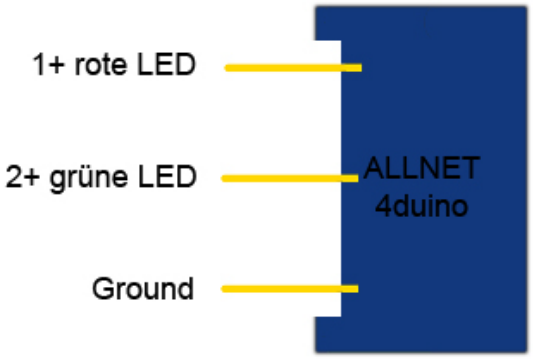
# B10/B11 Bicolor LED 5/3mm




## Brief Description / Technical Data

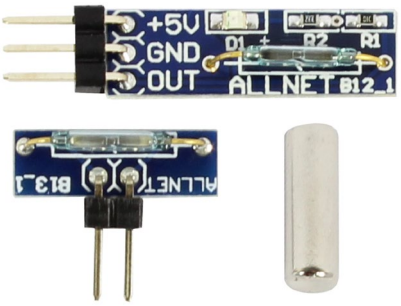
LED module containing a red and green LED. These are interconnected by a common cathode.

## Pin Assignment



## Code Sample

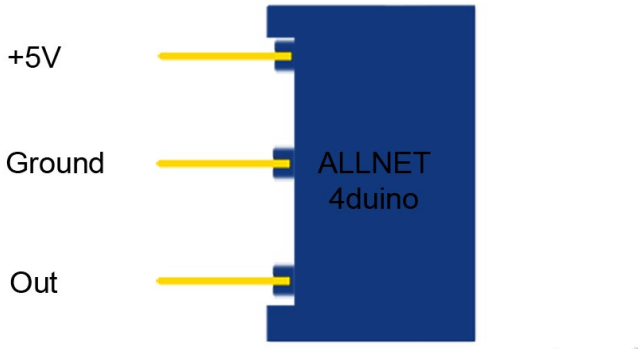
```
// ALLNET Bicolor LED 3/5mm B10/B11
// Information http://www.allnet.de

int B10_1 = 3;
int B10_2 = 2;
int B11_1 = 5;
int B11_2 = 4;

void setup() {
  Serial.begin(9600);
  pinMode(B10_1, OUTPUT);
  pinMode(B10_2, OUTPUT);
  pinMode(B11_1, OUTPUT);
  pinMode(B11_2, OUTPUT);
}

void loop() {
  digitalWrite(B10_1, HIGH);
  digitalWrite(B11_1, HIGH);
  delay(500);
  digitalWrite(B10_1, LOW);
  digitalWrite(B11_1, LOW);
  digitalWrite(B10_2, HIGH);
  digitalWrite(B11_2, HIGH);
  delay(500);
  digitalWrite(B10_2, LOW);
  digitalWrite(B11_2, LOW);
  delay(500);
  digitalWrite(B10_1, HIGH);
  digitalWrite(B11_1, HIGH);
  digitalWrite(B10_2, HIGH);
  digitalWrite(B11_2, HIGH);
  delay(500);
  digitalWrite(B10_1, LOW);
  digitalWrite(B11_1, LOW);
  digitalWrite(B10_2, LOW);
  digitalWrite(B11_2, LOW);
  delay(500);
}
```




## Brief Description / Technical Data

If a magnetic field is detected, the two pins are short-circuited or a signal is applied to the OUT pin.

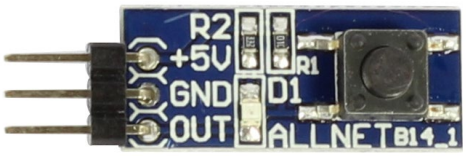
## Pin Assignment




## Code Sample

```
// ALLNET Reed Sensor B12/B13
// Information http://www.allnet.de


int pinSwitch = 2;
int pinLed = 3;
int StatoSwitch = 0;

void setup() {
  Serial.begin(9600);
  pinMode(pinLed, OUTPUT);
  pinMode(pinSwitch, INPUT);
}

void loop() {
  StatoSwitch = digitalRead(pinSwitch);
  if (StatoSwitch == HIGH) {
    digitalWrite(pinLed, HIGH);
    Serial.println("Switch : on");
  } else {
    digitalWrite(pinLed, LOW);
    Serial.println("Switch : off");
  }
}
```

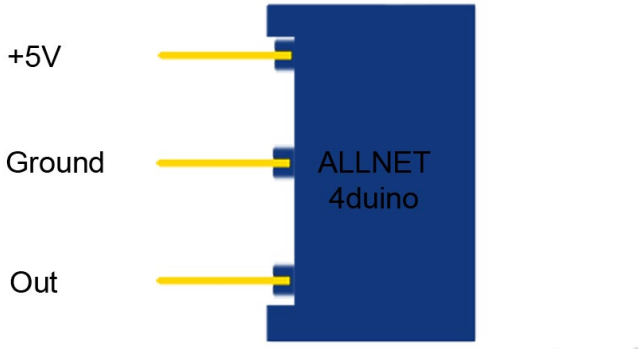
# B14 Button




## Brief Description / Technical Data

When pressing the button, a signal is output at the OUT pin.

## Pin Assignment




## Code Sample

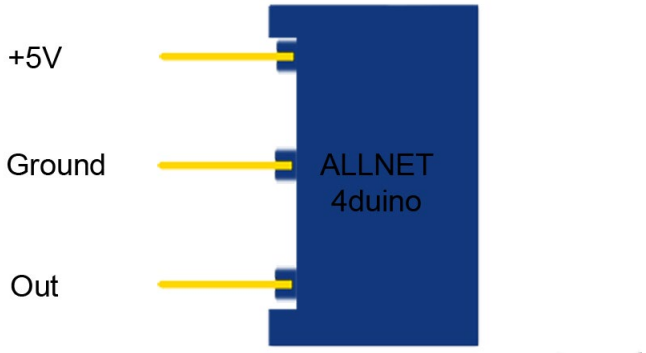
```
// ALLNET Button B14
// Information http://www.allnet.de


//Declaring the necessary variables
int Digital_Input = 3;

//one-time setup commands
void setup ()
{
  //Allocating pin function
  pinMode (Digital_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //When the value of Digital_Input equals 1
  if (digitalRead (Digital_Input) == 1)
  {
    //Then the button is pressed and this is output as message
    Serial.println ("Button pressed");
  }

  //Pause
  delay (200);
}
```

# B15 Tilt Sensor


## Brief Description / Technical Data

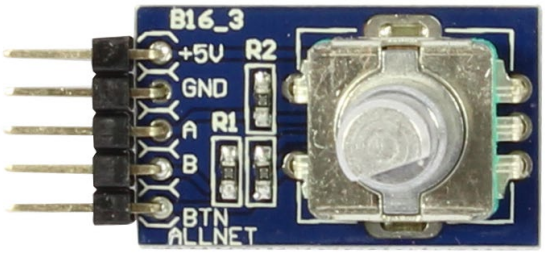
Depending on the inclination, a switch briefly short-circuits the input pins.

## Pin Assignment




## Code Sample

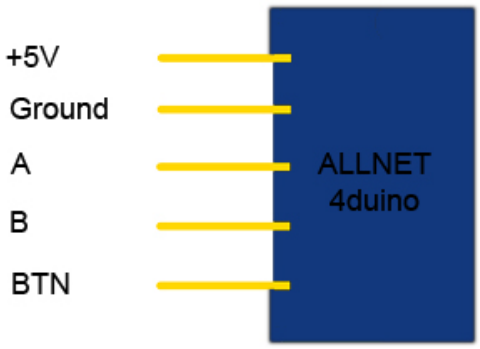
```
// ALLNET Tilt Sensor B15
// Information http://www.allnet.de


//Declaring the necessary variables
int Digital_Input = 3;

//one-time setup commands
void setup ()
{
  //Allocating pin function
  pinMode (Digital_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //When the value of Digital_Input equals 1
  if (digitalRead (Digital_Input) = 1)
  {
    //Then this is an incline and is output as message
    Serial.println ("Button pressed");
  }

  //Pause
  delay (200);
}
```




## Brief Description / Technical Data

In this circuit, a rotary encoder is read out by means of a pushbutton. The values are output via the serial interface.

## Pin Assignment



ALLNET 4duino

# Code Sample

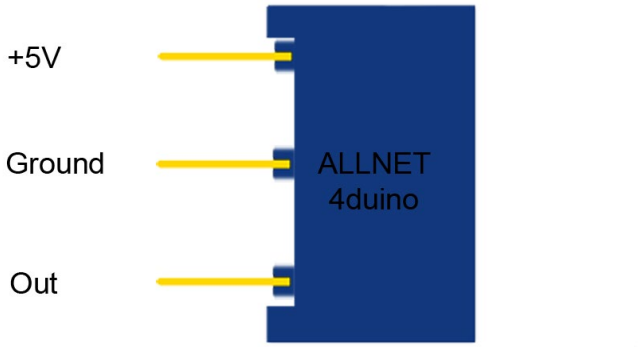
```
// ALLNET Rotary Encoder B16
// Information http://www.allnet.de


/* Read Quadrature Encoder
  * Connect Encoder to Pins encoder0PinA,
encoder0PinB, and +5V.
  */


 int val;
 int encoder0PinA = 7;
 int encoder0PinB = 6;
 int encoder0PinC = 5;
 int encoder0Pos = 0;
 int encoder0PinALast = LOW;
 int n = LOW;


 void setup() {
   pinMode (encoder0PinA,INPUT);
   pinMode (encoder0PinB,INPUT);
   pinMode (encoder0PinC,INPUT);
   Serial.begin (9600);
 }

 void loop() {
   n = digitalRead(encoder0PinA);
   if ((encoder0PinALast == LOW) && (n == HIGH)) {
     if (digitalRead(encoder0PinB) == LOW) {
       encoder0Pos--;
     } else {
       encoder0Pos++;
     }
        Serial.print ("Encoder: ");
   Serial.print (encoder0Pos);
   Serial.print (", ");
    int button = digitalRead(encoder0PinC);
   Serial.print ("Button pressed: ");
   Serial.print (button);
   Serial.println (" ");

   }


   encoder0PinALast = n;


 }
```

# B18 Light Barrier


## Brief Description / Technical Data

In this circuit a light barrier is used as a switch to turn an LED on/off.

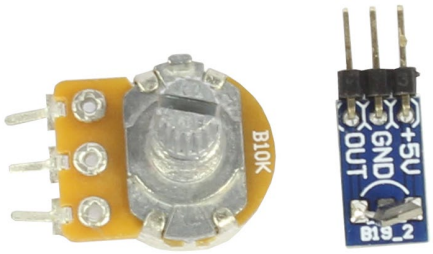
## Pin Assignment




## Code Sample

```
// ALLNET Tilt Light Barrier B18
// Information http://www.allnet.de


int sensorPin = 3;
int ledPin = 2;

void setup() {
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  int val = digitalRead(sensorPin);
  Serial.println(val);
  if ( val == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

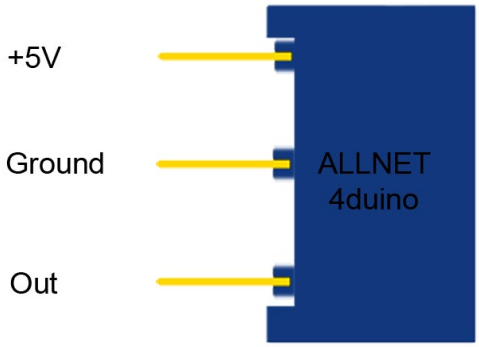
# B19 Potentiometer / Analog Hall




## Brief Description / Technical Data

The resistance of the potentiometer changes according to the position of the rotary switch. This changes the voltage applied to the sensor pin. The analog hall sensor works in the same way. Here, however, the resistance changes according to the magnetic environment.

## Pin Assignment



ALLNET 4duino

# Code Sample

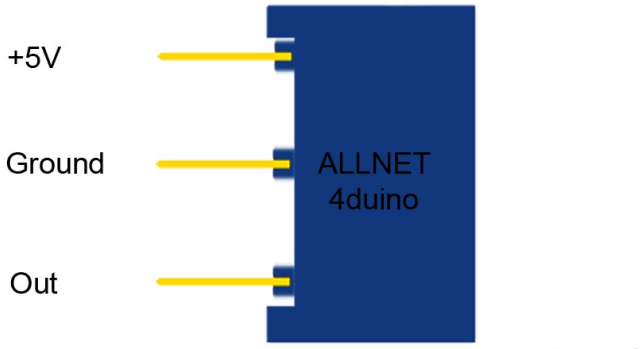
```
// ALLNET Potentiometer/Analog Hall B19
// Information http://www.allnet.de

//Declaring the necessary variables
int Analog_Input = A0;

//one-time setup commands
void setup ()
{
  //Allocating pin function
  pinMode (Analog_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //Declare a temporary buffer
  float analog;

  //Current value is read, converted to voltage value...
  analog = analogRead (Analog_Input) * (5.0 / 1023.0);

  //... and output here
  Serial.print ("Analog voltage value:");
  Serial.print (analog);
  Serial.print (" V, ");
  Serial.println ("---------------------------------");

  //Pause
  delay (200);
}
```

# B20 Temperature Sensor Module


## Brief Description / Technical Data

Communication protocol: 1-wire.
Provides a 9-12 bit accurate temperature measurement via the 1-wire pin.

## Pin Assignment



## Code Sample

Two additional libraries are required for the following code sample:

- [OneWire Library]
- [Dallas Temperature Control Library]

Both libraries can be installed directly in the Arduino IDE. They are available for download in the Library Manager.

**Pin assignment:**

Sensor +V = [Pin 5V]
Sensor GND = [Pin GND]
Sensor DQ = [Pin Out]

```
// ALLNET Temperature Sensor Module B20
// Information http://www.allnet.de


//Necessary libraries are imported
#include <DallasTemperature.h>
#include <OneWire.h>

//Here the input pin is declared, to which the sensor
module is connected
#define B20_Signal_PIN 4

//Libraries are configured
OneWire oneWire(B20_Signal_PIN);
DallasTemperature sensors(&oneWire);

//one-time setup commands
void setup() {

  //Start serial transmission
  Serial.begin(9600);
  Serial.println("B20 Temperature measurement");

  //Sensor is initialized
  sensors.begin();
}

//permanently repeated main loop
void loop()
{
  //Temperature measuring is started...
  sensors.requestTemperatures();

  //... and measured temperature is output
  Serial.print("Temperature: ");
  Serial.print(sensors.getTempCByIndex(0));
  Serial.write(176); // UniCode specification of char symbol for "° symbol"
  Serial.println("C");

  //Pause
  delay (1000);
}
```

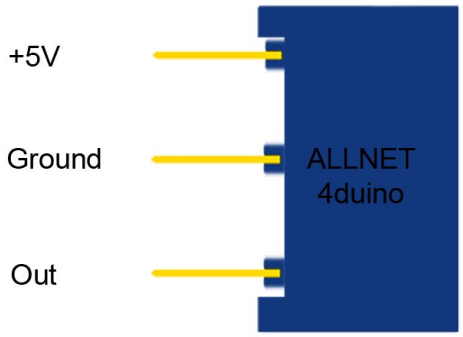
# B21 IR LED

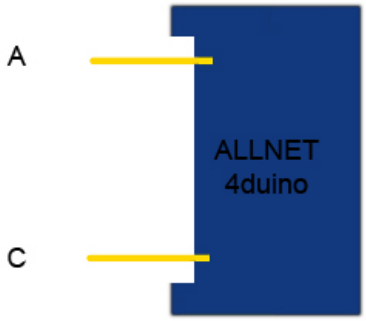


## Brief Description / Technical Data

A light emitting diode emitting in the infrared range.

## Pin Assignment



# Code Sample

By means of the two sensor modules B21 and B22, an infrared remote control + infrared receiver system can be set up. In addition to the two modules, two individual Arduinos are required for this purpose. One of them functions as a transmitter and the other receives the signals and outputs them in the serial console.

The following code sample requires an additional library:

- [Arduino-IRRemote]

The library can be installed directly in the Arduino IDE. It can be downloaded from Library Manager.

In infrared transmission systems, there are different protocols in which the data can be sent. In the following example the RC5 protocol is used for sending - the library "Arduino-IRremote" takes care of the conversion to the correct data sequence. However, there are also other protocols/encodings within the library - these are marked in the documentation/code of the library.

Code for receiver (B22):

**Pin assignment:**

| | |
|---|---|
| Sensor V+ | = [Pin 5V] |
| Sensor Signal | = [Pin 11] |
| Sensor GND | = [Pin GND] |

```
// ALLNET IR LED B21
// Information http://www.allnet.de

//Arduino IR remote library is added
#include <IRremote.h>

//Declaring the necessary variables
int RECV_PIN = 11;

// Arduino IR remote library is initialized
IRrecv irrecv(RECV_PIN);
decode_results results;

//one-time setup commands
void setup()
{
//Start serial transmission
  Serial.begin(9600);

  //Infrared receiver is started
  irrecv.enableIRIn();
}

//permanently repeated main loop
void loop()
{
  //When a signal has been received on the receiver
  if (irrecv.decode(&results))
  {
    //Then output the received in serial console
    Serial.println(results.value, HEX);

    //Continue receiving
    irrecv.resume();
  }
}
```

Code for transmitter (B21):

**Pin assignment:**

LED Signal = [Pin 3]
LED GND = [Pin GND]

```
// ALLNET IR LED B21
// Information http://www.allnet.de

//Arduino IR remote library is added...
#include <IRremote.h>

//...and here initialized
IRsend irsend;

// Output settings are adopted from library
// Respective outputs differ according to used Arduino
// Arduino UNO: output = D3
// Arduino MEGA: output = D9
// Find complete list of respective outputs at // http://z3t0.github.io/Arduino-IRremote/

//one-time setup commands
void setup()
{
}

//permanently repeated main loop
void loop()
{
  // In this example, the transmitter sends signal A90 (in hexadecimal format) in coding
"RC5"
  // This is transmitted three times successively, followed by a 5-second pause
  for (int i = 0; i < 3; i++)
  {
    // [0xA90] signal to be sent | [12] bit length of signal to be sent (hex A90
= 1010 1001 0000)
    irsend.sendRC5(0xA90, 12);
    delay(40);
  }
}
```

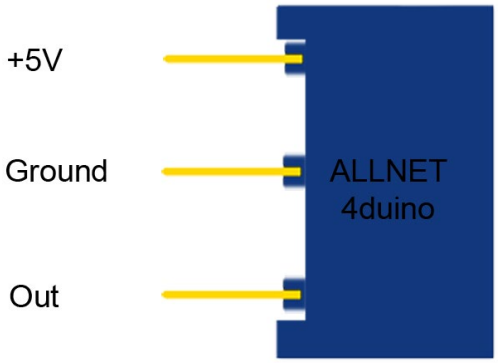
# B22 IR Receiver 38kHz


## Brief Description / Technical Data

Carrier frequency: 38kHz. Can receive infrared signals and outputs them as a digital sequence at the signal output.

In addition, the LED integrated on the module flashes briefly when an infrared signal has been detected.

## Pin Assignment



# Code Sample

By means of the two sensor modules B21 and B22, an infrared remote control + infrared receiver system can be set up. In addition to the two modules, two individual Arduinos are required for this purpose. One of them functions as a transmitter and the other receives the signals and outputs them in the serial console.

The following code sample requires an additional library:

- [Arduino-IRremote]

The library can be installed directly in the Arduino IDE. It can be downloaded from Library Manager.

In infrared transmission systems, there are different protocols in which the data can be sent. In the following example the RC5 protocol is used for sending - the library "Arduino-IRremote" takes care of the conversion to the correct data sequence. However, there are also other protocols/encodings within the library - these are marked in the documentation/code of the library.

Code for receiver (B22):

**Pin assignment:**

| | |
|---|---|
| Sensor V+ | = [Pin 5V] |
| Sensor Ground | = [Pin Ground] |
| Sensor Out | = [Pin Digital] |

```
//Start serial transmission
  Serial.begin(9600);

  //Infrared receiver is started
  irrecv.enableIRIn();
}

//permanently repeated main loop
void loop()
{
  //When a signal has been received on the receiver
  if (irrecv.decode(&results))
  {
    //Then output the received signal to the serial console
    Serial.println(results.value, HEX);

    //Continue receiving
    irrecv.resume();
  }
}
```

ALLNET 4duino

```
// ALLNET IR Reciever 38mHz B22
// Information http://www.allnet.de

//Arduino IR remote library is added
#include <IRremote.h>

//Declaring the necessary variables
int RECV_PIN = 11;

// Arduino IR remote library is initialized
IRrecv irrecv(RECV_PIN);
decode_results results;
//one-time setup commands
void setup()
{

  //Start serial transmission
  Serial.begin(9600);

  //Infrared receiver is started
  irrecv.enableIRIn();
}

//permanently repeated main loop
void loop()
{
  //When a signal has been received on the receiver
  if (irrecv.decode(&results))
  {
    //Then output the received signal to the serial console
    Serial.println(results.value, HEX);

    //Continue receiving
    irrecv.resume();
  }
}
```

Code for transmitter (B21):

**Pin assignment:**

LED +V = [Pin 3]
LED GND = [Pin GND]

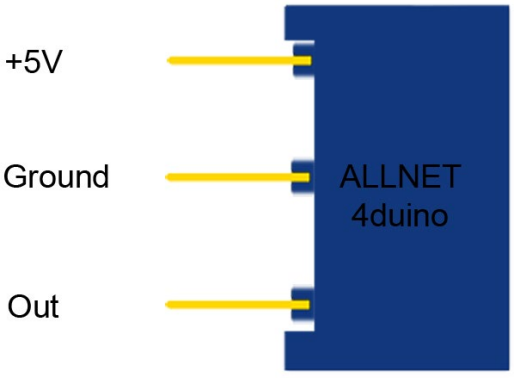
```
// ALLNET IR Reciever 38mHz B22
// Information http://www.allnet.de

//Arduino IR remote library is added...
#include <IRremote.h>

//...and here initialized
IRsend irsend;

// Output settings are adopted from library
// Respective outputs differ according to used Arduino
// Arduino UNO: Output = D3
// Arduino MEGA: Output = D9
// Find complete list of respective outputs at // http://z3t0.github.io/Arduino-IRremote/

//one-time setup commands
void setup()
{
}

//permanently repeated main loop
void loop()
{
  // In this example, the transmitter sends signal A90 (in hexadecimal format) in coding
"RC5"
  // This is transmitted three times successively, followed by a 5-second pause
  for (int i = 0; i < 3; i++)
  {
    // [0xA90] signal to be sent | [12] bit length of signal to be sent (hex A90
= 1010 1001 0000)
    irsend.sendRC5(0xA90, 12);
    delay(40);
  }
}
```

# B23 Shock Sensor


## Brief Description / Technical Data

The sensor pin emits a signal as soon as vibration is detected.

## Pin Assignment



# Code Sample

This is an example program which triggers a serial output when vibration is detected.

**Pin assignment:**

| | |
|---|---|
| Sensor +V | = [Pin 5V] |
| Sensor GND | = [Pin GND] |
| Sensor Digital | = [Pin 3] |

```
// ALLNET Tilt Sensor B15
// Information http://www.allnet.de


//Declaring the necessary variables
int Digital_Input = 3;

//one-time setup commands
void setup ()
{
  //Allocate pin function
  pinMode (Digital_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //When the value of Digital_Input equals 1
  if (digitalRead (Digital_Input) = 1)
  {
    //Then vibration is detected and this is output
as message
    Serial.println ("Vibration detected");
  }

  //Pause
  delay (200);
}
```

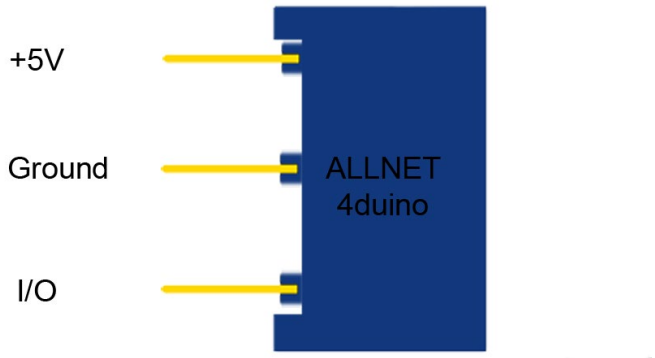
# B24 Temperature & Humidity


## Brief Description / Technical Data

Chipset: DHT11 | Communication protocol: 1-wire. Measuring range humidity: 20 - 90% RH. Measuring range temperature: 0 - 50°C

Advantages of this sensor are the combination of temperature measurement and humidity measurement in a compact design - the disadvantage is the low sampling rate of the measurement, so that only every 2 seconds a new measurement result is available - this sensor is therefore very well suited for long-term measurements.

## Pin Assignment



# Code Sample

This is an example program, which outputs the sensor value to the serial monitor.

**Pin assignment:**

| | |
|---|---|
| Sensor +V | = [Pin 5V] |
| Sensor GND | = [Pin GND] |
| Sensor Digital | = [Pin 2] |

```
// ALLNET Temperature & Humidity B24
// Information http://www.allnet.de


//Arduino DHT library is added
#include "DHT.h"

// Arduino DHT library is initialized
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

//one-time setup commands
void setup()
{
  //Start serial transmission
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  //DHT11 Sensor is started
  dht.begin();
}

void loop()
{
  //Pause
  delay(2000);

  //Read humidity
  float h = dht.readHumidity();
```

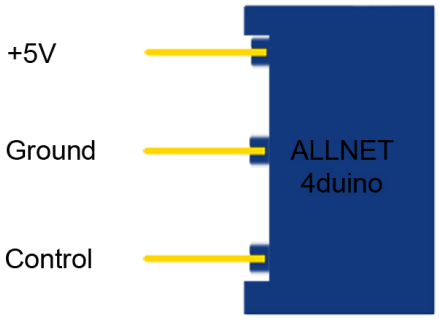
```
//Read temperature in Celsius
  float t = dht.readTemperature();
  //Read temperatur in Farenheit
  float f = dht.readTemperature(true);

  //Checks whether a measurement is faulty and in
this case aborts prematurely
  //Respective information is provided in serial
```

# B25 1 Watt LED Module


## Brief Description / Technical Data

The module has a 1W LED, which can be switched on/off by the control pin.

## Pin Assignment



# Code Sample

This is an example program, which allows the 1W LED to flash.

**Pin assignment:**

| | |
|---|---|
| Sensor +V | = [Pin 5V] |
| Sensor GND | = [Pin GND] |
| Sensor Control | = [Pin 3] |

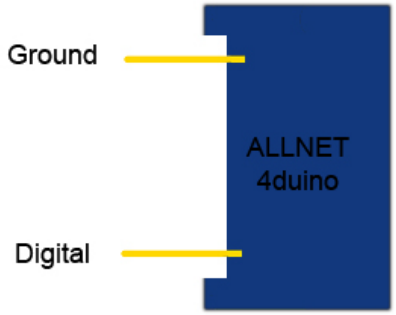
```
// ALLNET 1W LED Module B25
// Information http://www.allnet.de

//Declaring the necessary variables
int LED_Input = 3;

//one-time setup commands
void setup()
{
  //Allocate pin function
  pinMode (LED_Input, INPUT);
}

//permanently repeated main loop
void loop()
{
  //LED on
  digitalWrite (LED_Input, HIGH);

  //Pause
  delay(1000);

  //LED off
  digitalWrite (LED_Input, LOW);

  //Pause
  delay(1000);
}
```

# B26 Piezo Speaker


## Brief Description / Technical Data

The piezo speaker emits a tone if power is applied to the pins.

## Pin Assignment



# Code Sample

This is an example program, which sends an alarm signal.

**Pin assignment:**

| | |
|---|---|
| Sensor GND | = [Pin GND] |
| Sensor Digital | = [Pin 8] |

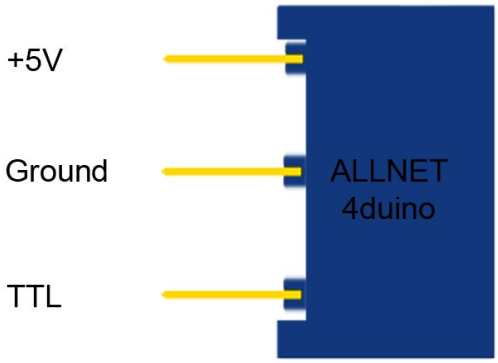
```
// ALLNET Piezo Speaker B26
// Information http://www.allnet.de


//one-time setup commands
void setup()
{
}

//permanently repeated main loop
void loop()
{
  //Activate piezo on pin 8 with tone 100
  tone(8,100);

  //Pause
  delay(1000);

  //Deactivate piezo on pin 8
  noTone(8);

  //Pause
  delay(1000);
}
```

# B27 Buzzer


## Brief Description / Technical Data

The buzzer emits a tone if a signal is present at the TTL pin.

## Pin Assignment



# Code Sample

This is an example program, which sends an alarm signal.

**Pin assignment:**

| | |
|---|---|
| Sensor +5V | = [+5V] |
| Sensor GND | = [GND] |
| Sensor Digital | = [TTL] |

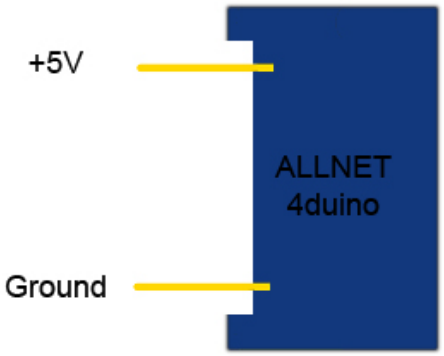
```
// ALLNET Buzzer B27
// Information http://www.allnet.de

//Declaring the necessary variables
int Buzzer_Input = 3;

//one-time setup commands
void setup()
{
  //Allocate pin function
  pinMode (Buzzer_Input, INPUT);
}

//permanently repeated main loop
void loop()
{
  //Buzzer on
  digitalWrite (Buzzer_Input, HIGH);

  //Pause
  delay(500);

  //Buzzer off
  digitalWrite (Buzzer_Input, LOW);

  //Pause
  delay(500);
}
```

# B28 Flash LED


## Brief Description / Technical Data

The LED starts flashing if voltage is applied between the two pins.

## Pin Assignment



# Code Sample

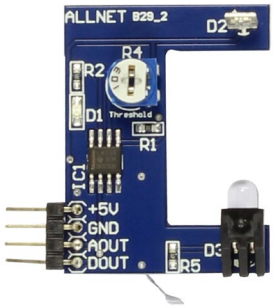
This is an example program, which imitates a flashing light.

**Pin assignment:**

| | |
|---|---|
| Sensor +5V | = [+5V] |
| Sensor Ground | = [Ground] |

```
// ALLNET Flash LED B28
// Information http://www.allnet.de

//Declaring the necessary variables
int LED_Input = 3;

//one-time setup commands
void setup()
{
  //Allocate pin function
  pinMode (LED_Input, INPUT);
}

//permanently repeated main loop
void loop()
{
  //LED flashes
  digitalWrite (LED_Input, HIGH);

  //Pause
  delay(500);

  //LED off
  digitalWrite (LED_Input, LOW);

  //Pause
  delay(500);
}
```

# B29 Heartbeat




## Brief Description / Technical Data

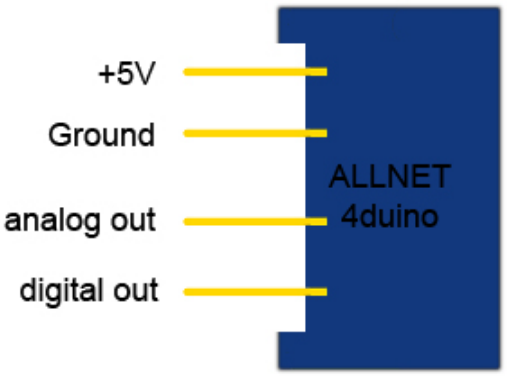
Digital output: If a heartbeat is detected, a signal is output here

Analog output: Direct measured value of the sensor unit.

LED1: Indicates that the sensor is powered.

LED2: Indicates that the heart rate has exceeded the limit.

## Pin Assignment



## Code Sample

The program reads out the current voltage value, which can be measured at the analog output and outputs it on the serial interface.

In addition, the status of the digital pin in the console is also indicated, which means whether or not the limit has fallen below the limit value.

**Pin assignment:**

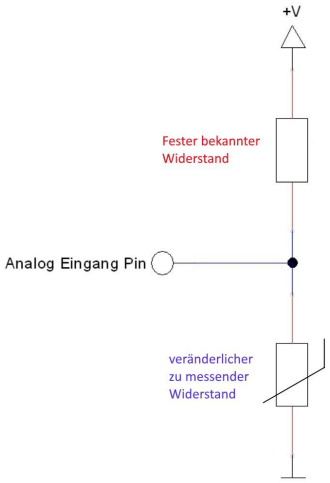
| | |
|---|---|
| Sensor +V | = [Pin +5V] |
| Sensor GND | = [Pin GND] |
| Sensor Digital | = [Pin 4] |
| Sensor Analog | = [Pin 3] |

```
// ALLNET Heartbeat B29
// Information http://www.allnet.de


//Declaring the necessary variables
int Analog_Input = A0;
int Digital_Input = 3;

//one-time setup commands
void setup ()
{
  //Allocate pin function
  pinMode (Analog_Input, INPUT);
  pinMode (Digital_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //Declare temporary buffers
  float analog;
  int digital;

//Current values are read, converted to voltage values...
  analog = analogRead (Analog_Input);
  digital = digitalRead (Digital_Input);

  //... and output here
  Serial.print ("Analog voltage value:");
  Serial.print (analog);
  Serial.print (", ");
  Serial.print ("Threshold:");

  //When the value equals digital 1
  if (digital = 1)
  {
    //Then the threshold is reached and this is output as message
    Serial.println (" reached");
  }
  else
  {
    //Otherwise the threshold is not reached and this is output as message
    Serial.println (" not yet reached");
  }

  // Optical separation of data in serial output
  Serial.println ("---------------------------------");

  //Pause
  delay (200);
}
```

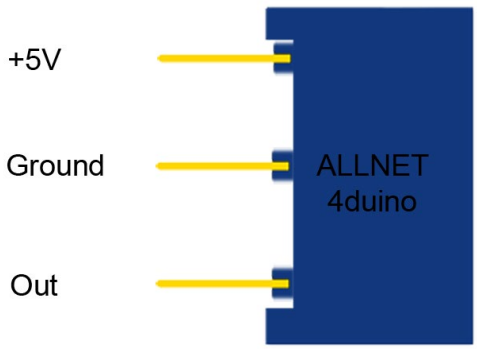

## Brief Description / Technical Data

Contains an LDR resistor whose resistance value decreases in brighter environments.

This resistance can be determined with the help of a voltage divider, where a known voltage is divided by a known and an unknown (changeable) resistance. The resistance can then be calculated by means of this measured voltage - the exact calculation is included in the code samples below.




## Pin Assignment

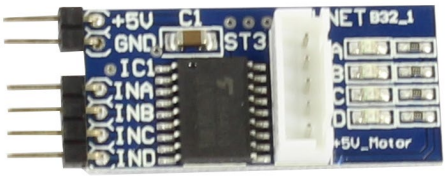



## Code Sample

The program measures the current voltage value at the sensor, calculates the actual resistance value of the sensor from this and the known series resistance and outputs the results on the serial output.

**Pin assignment:**
Sensor V+ = [Pin +5V]
Sensor GND = [Pin GND]
Sensor Analog = [Pin A0]

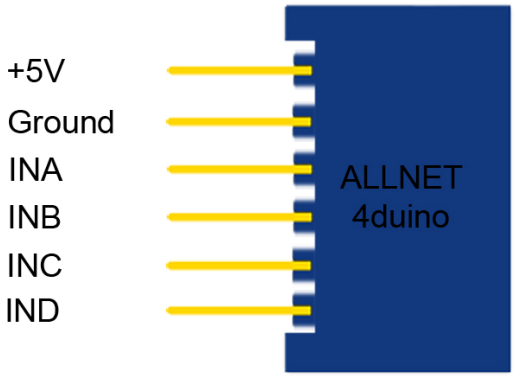
```
// ALLNET Photoresistor B30
// Information http://www.allnet.de

//Declaring the necessary variables
int sensorPin = A0;

//one-time setup commands
void setup()
{
  //Allocating pin function
  pinMode (sensorPin, INPUT);

  //Start serial transmission
  Serial.begin(9600);
}

// The program measures the current voltage value at the sensor,
// calculates from this and the known series resistance the the current
// resistance value of the sensor and outputs the results on the serial output

void loop()
{
  // Current value is measured and voltage and resistance value are calculated...
  int rawValue = analogRead(sensorPin);
  float voltage = rawValue * (5.0 / 1023) * 1000;
  float resitance = 10000 * ( voltage / ( 5000.0 - voltage) );

  // ... and here output to the serial interface
  Serial.print("Volltage value:");
  Serial.print(voltage);
  Serial.print("mV");
  Serial.print(", resistance value:");
  Serial.print(resitance);
  Serial.println("ohms");
  Serial.println("---------------------------------------");

  //Pause
  delay(500);
}
```




## Brief Description / Technical Data

The sensor enables the stepper motor to be operated with the Arduino.

## Pin Assignment



## Code Sample

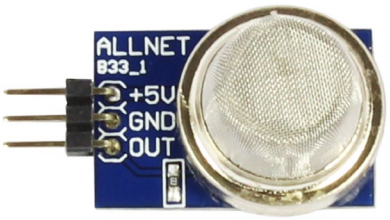
This is an example program, which lets a stepper motor slowly turn a step further.

**Pin assignment:**

| | | | |
|---|---|---|---|
| Sensor GND | = [Pin GND] | Sensor INB | = [Pin 9] |
| Sensor +5V | = [Pin +5V] | Sensor INC | = [Pin 10] |
| Sensor INA | = [Pin 8] | Sensor IND | = [Pin 11] |

```
// ALLNET 5V Stepper-Engline with PCB Driver B32
// Information http://www.allnet.de

//Necessary libraries are imported
#include <Stepper.h>

//Declaring the necessary variables
const int stepsPerRevolution = 32;
int stepCount = 0;

//Libraries are configured
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

//one-time setup commands
void setup()
{
  //Start serial transmission
  Serial.begin(9600);
}

//permanently repeated main loop
void loop()
{
  //Stepper motor 1 step
  myStepper.step(1);
  stepCount++;

  //Output of steps performed
  Serial.print("steps:");
  Serial.println(stepCount);

  //Pause
  delay(1000);
}
```

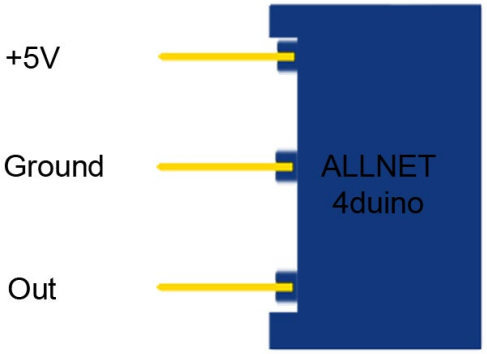
# B33 Gas MQ2 Sensor




## Brief Description / Technical Data

The sensor applies different voltages to the OUT pin, depending on the amount of gas in the air.

## Pin Assignment



## Code Sample

This is an example program, which outputs the analog sensor value on the serial display.

**Pin assignment:**

| | |
|---|---|
| Sensor GND | = [Pin +5V] |
| Sensor +5V | = [Pin Ground] |
| Sensor Output | = [Pin A0] |

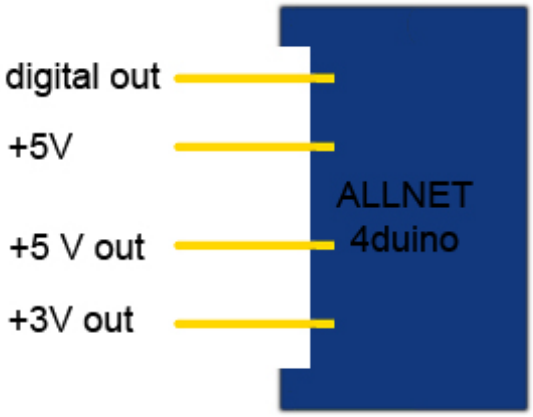
```
// ALLNET Gas MQ2 Sensor B33
// Information http://www.allnet.de

//Declaring the necessary variables
int sensorPin = A0;

//one-time setup commands
void setup()
{
  //Allocate pin function
  pinMode(sensorPin, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop()
{
    //Temporary variable with sensor value
    int val = analogRead (sensorPin);

    //Output sensor value
    Serial.print ("Sensor value:  ");
    Serial.println (val);
    Serial.println ("---------------------------------------");

    //Pause
    delay (1000);
}
```

# B34 Voltage Regulator linear

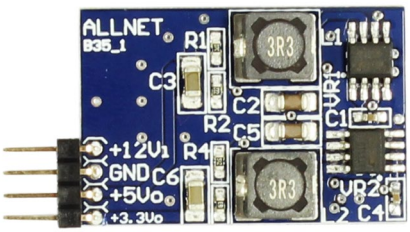

## Brief Description / Technical Data

The ALLNET B34 is a linear voltage regulator. It lowers an applied 12V voltage to 5 and 3 volts. If the input voltage is changed, the output voltages will change in the same way.

## Pin Assignment



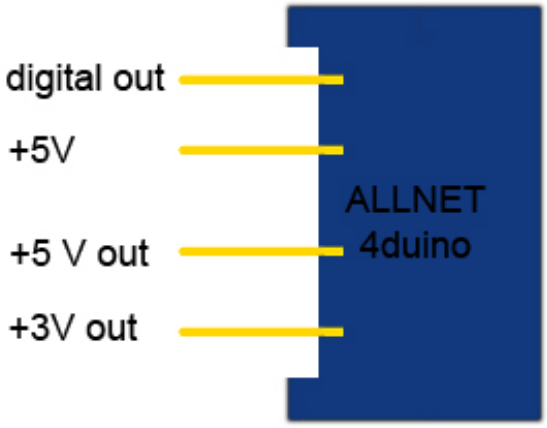
# B35 Voltage Regulator




## Brief Description / Technical Data

The ALLNET B35 is a voltage regulator. It lowers an applied 12V voltage to 5 and 3 volts.

## Pin Assignment

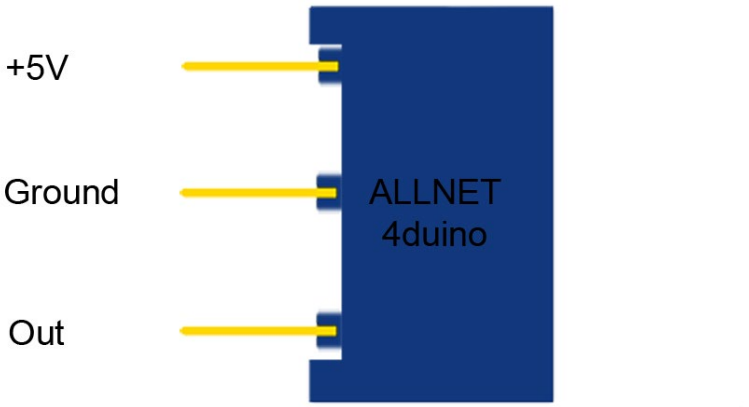



## B36 Motion Detection / Bewegungsmelder


## Brief Description / Technical Data

The motion detector outputs a signal when movement is detected.

## Pin Assignment




## Code Sample

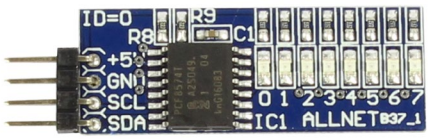
This is an example program, which outputs movement to the serial monitor.

**Pin assignment:**

| | |
|---|---|
| Sensor GND | = [Pin GND] |
| Sensor +5V | = [Pin 5V] |
| Sensor Output | = [Pin 3] |

```
// ALLNET Motion Detection B36
// Information http://www.allnet.de

//Declaring the necessary variables
int Motion_Input = 3;

//one-time setup commands
void setup ()
{
  //Allocate pin function
  pinMode (Motion_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //When the value of Digital_Input equals 1
  if (digitalRead (Motion_Input) = 1)
  {
    //Then movement is detected and this is output as message
    Serial.println ("Movement detected");
  }

  //Pause
  delay (200);
}
```

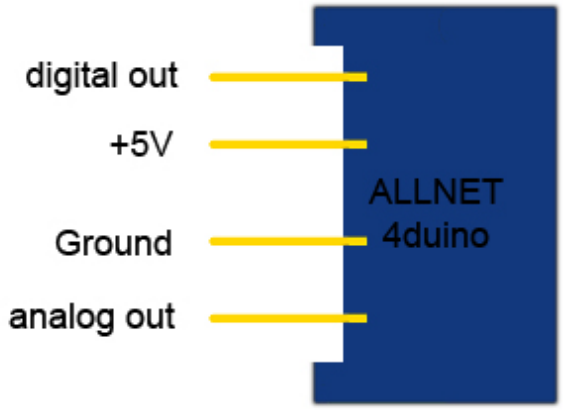
# B37 8 LED PCB




## Brief Description / Technical Data

The 8-LED PCB allows 8 separate LEDs to be controlled via the $I^2C$ bus.

## Pin Assignment



# Code Sample

The sample program simulates dropping battery indication

**Pin assignment:**

| | |
|---|---|
| Sensor GND | = [Pin GND] |
| Sensor +5V | = [Pin 5V] |
| Sensor Output | = [Pin A0] |

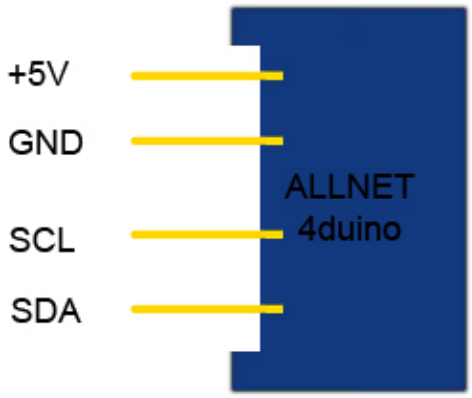
```
// ALLNET Flash LED B28
// Information http://www.allnet.de

//Necessary libraries are imported
#include<Wire.h>

//Declaring the necessary variables
int counter = 0;
int power = 7 ;
int y = 128;

//one-time setup commands
void setup()
{
  //Start I2C sensor connection
  Wire.begin();

  //Start serial transmission
  Serial.begin (9600); // 9600 bps
}

//permanently repeated main loop
void loop()
{
  if (counter % 10000 == 0)
  {
    //Serial output of variables
    Serial.print (power, DEC);
    Serial.print (" - ");
    Serial.println (y, DEC);
    //Start data transmission
    Wire.beginTransmission(32);
    //Send data
    Wire.write(y);
    //End data transmission
    Wire.endTransmission();
    y = y / 2;
    power = power - 1;

  }
  counter++;
  if (power < 0)
  {
    y = 128;
    power = 7;
  }

  //Pause
  delay(1);
}
```

# B38 Temperature I2C Sensor


## Brief Description / Technical Data

The ALLNET B38 is a digital temperature sensor that communicates via I²C.

## Pin Assignment



# Code Sample

This is an example program, which outputs the analog sensor value on the serial display.

**Pin assignment:**

| | |
|---|---|
| Sensor GND | = [Pin GND] |
| Sensor +5V | = [Pin 5V] |
| Sensor SCL | = [Pin A5] |
| Sensor SDA | = [Pin A4] |

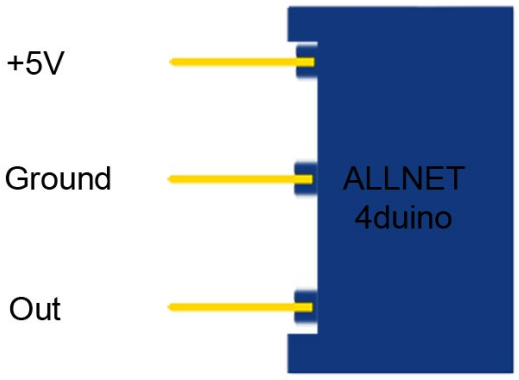
```
// ALLNET Temperature I2C Sensor B38
// Information http://www.allnet.de


//Necessary libraries are imported
#include <Wire.h>

//Declaring the necessary variables
#define adress 0x4F

//one-time setup commands
void setup()
{
  //Start serial transmission
  Serial.begin(9600);

  //Start sensor connection
  Wire.begin();
}

//permanently repeated main loop
void loop()
{
  //Execute below declared function
  int c1 = read_temp(address);

  // Output determined sensor value
  Serial.print("Sensor 1: ");
  Serial.print(c1);

  //Pause
  delay(500);
}

int read_temp(int address)
{
  //Start transmission with sensor
  Wire.beginTransmission(address);
  //Send bit to receive information
  Wire.write(0x00);
  //Request 1 Byte from sensor
  Wire.requestFrom(address, 2);
  //Wait for response
  if (Wire.available() == 0)
  {
  //Save and return the value
  int c = Wire.read();
  }
  //End transmission
  Wire.endTransmission();
  return c;
}
```

# B39 Vibration Sensor


## Brief Description / Technical Data

The sensor applies a signal to the OUT Pin when vibration is detected.

## Pin Assignment



# Code Sample

This is an example program, which outputs the analog sensor value on the serial display.

**Pin assignment:**

| | |
|---|---|
| Sensor GND | = [Pin GND] |
| Sensor +5V | = [Pin 5V] |
| Sensor Output | = [Pin 3] |

```
// ALLNET Vibration Sensor B39
// Information http://www.allnet.de

//Declaring the necessary variables
int Vibration_Input = 3;

//one-time setup commands
void setup ()
{
  //Allocate pin function
  pinMode (Vibration_Input, INPUT);

  //Start serial transmission
  Serial.begin (9600);
}

//permanently repeated main loop
void loop ()
{
  //When the value of Digital_Input equals 1
  if (digitalRead (Vibration_Input) == 1)
  {
    //Then vibration is detected and this is output as message
    Serial.println ("Vibration detected");
  }

  //Pause
  delay (1000);
}
```