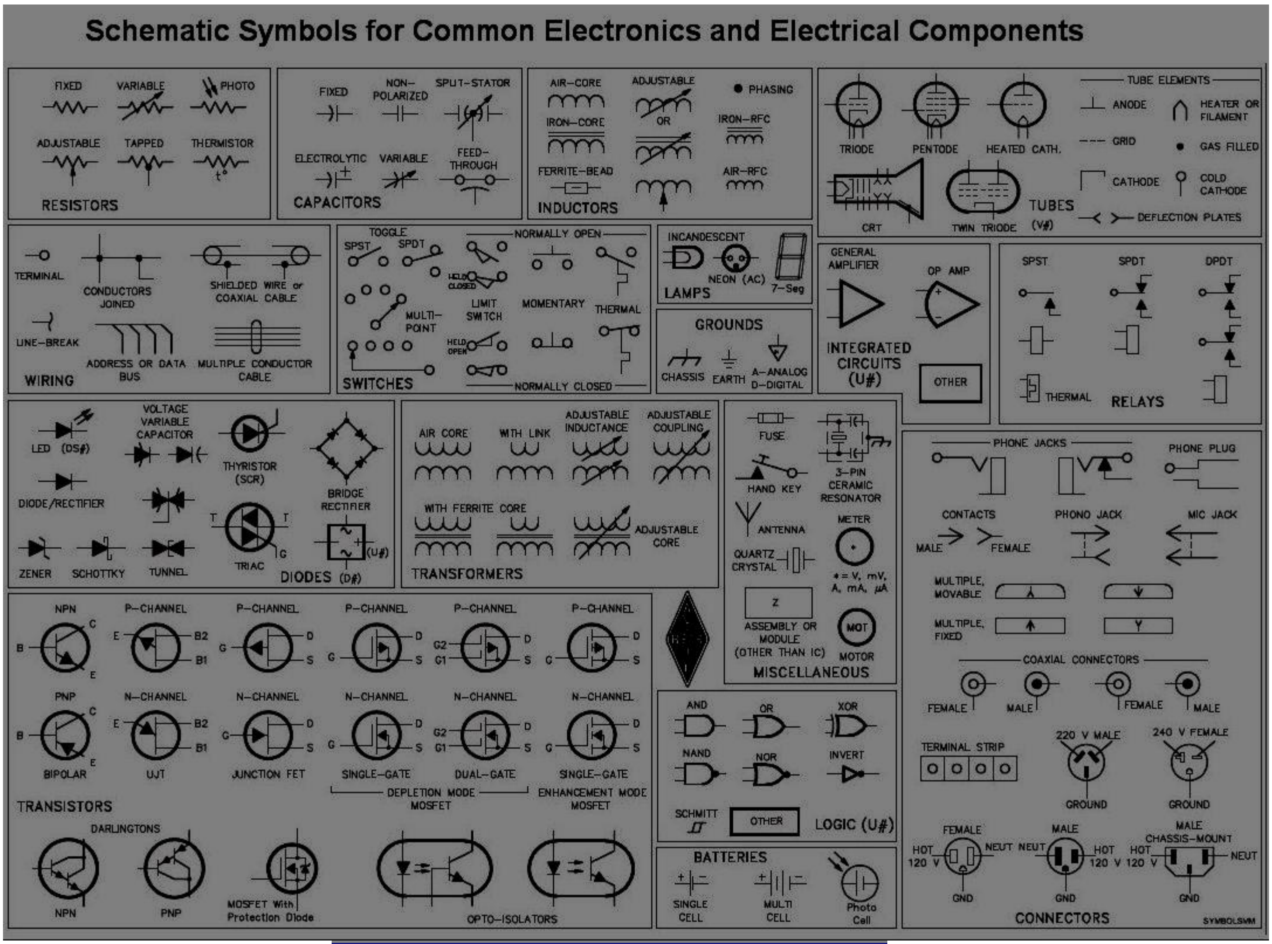
The Camera Remote: Fotomind MINI

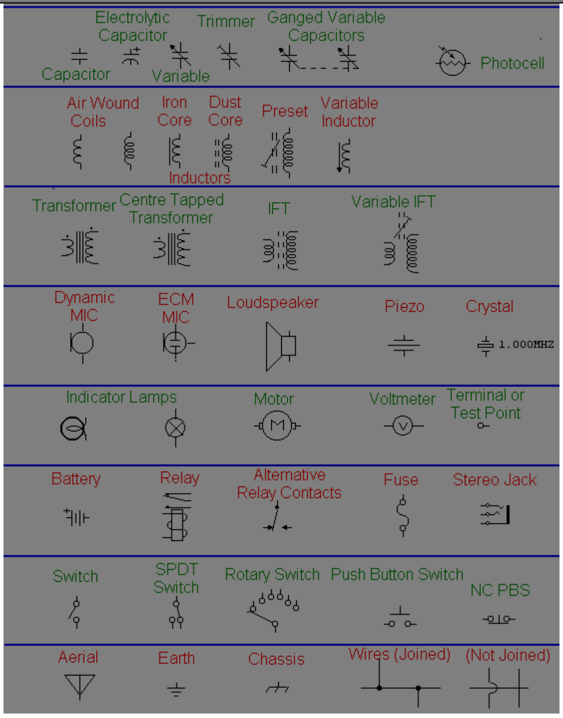
by DI Chrisitan Munk

☞ DI Y

Scematic Symbols:

Source: http://atmega32-avr.com/circuit-schematic-symbols/



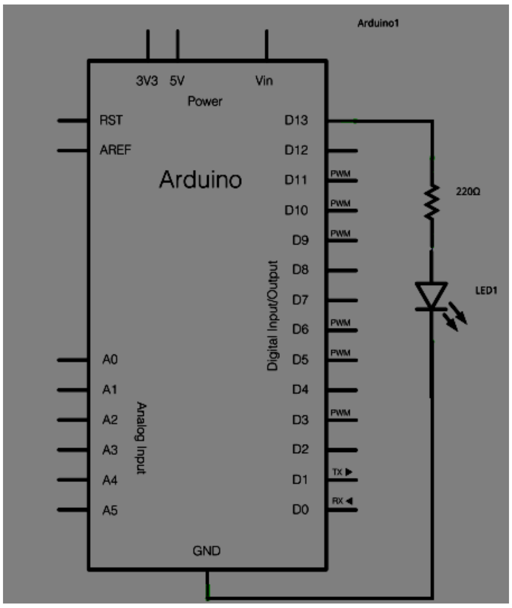


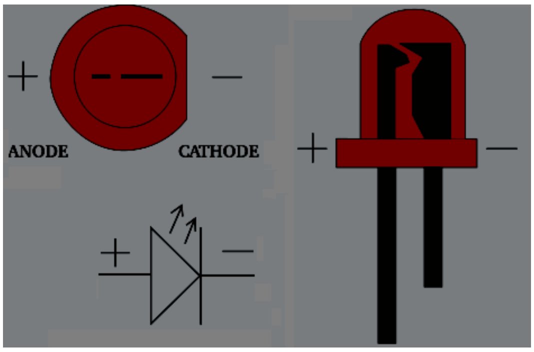
Source: http://arduino.cc/en/Tutorial/

Basic Programms:

LED (digital Output) :

Scematic:



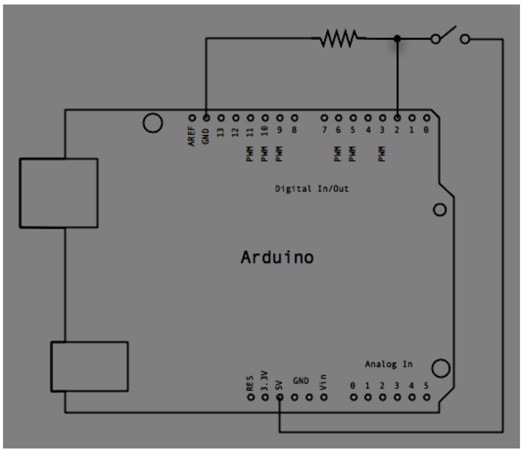


Code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// 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
}
```

Button (digital Input):

Scematic:

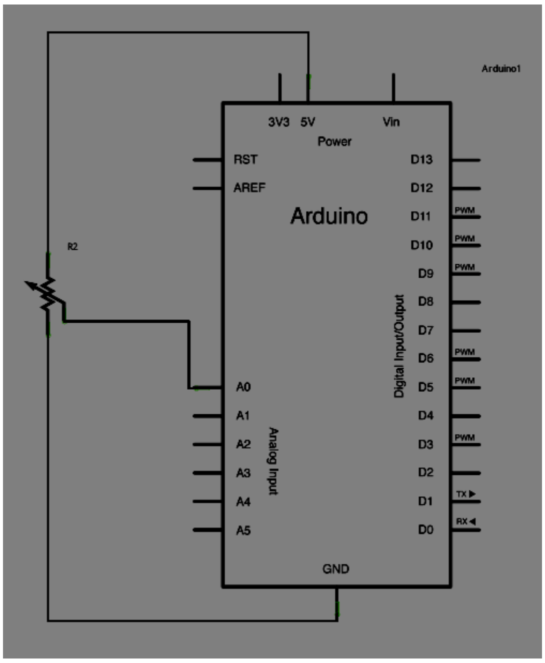


Code:

```
/*
 Turns on and off a light emitting diode(LED) connected to digital
 pin 13, when pressing a pushbutton attached to pin 2.
 The circuit:
 * LED attached from pin 13 to ground
 * pushbutton attached to pin 2 from +5V
 * 10K resistor attached to pin 2 from ground
 */
const int buttonPin = 2;      // the number of the pushbutton pin
const int ledPin =   13;      // the number of the LED pin

// variables will change:
int buttonState = 0;          // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }}
```

Potentiometer: (analog Input)

Scematic:



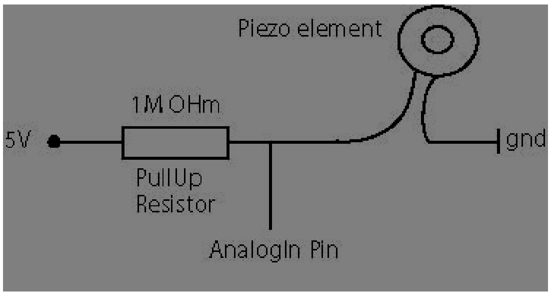
Code:

```
/* Analog Read to LED
 * ------------------
 *
 * turns on and off a light emitting diode(LED) connected to digital
 * pin 13. The amount of time the LED will be on and off depends on
 * the value obtained by analogRead(). In the easiest case we connect
 * a potentiometer to analog pin 2.
 */

int potPin = 2;    // select the input pin for the potentiometer
int ledPin = 13;   // select the pin for the LED
int val = 0;       // variable to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT);  // declare the ledPin as an OUTPUT
}

void loop() {
  val = analogRead(potPin);    // read the value from the sensor
  digitalWrite(ledPin, HIGH);  // turn the ledPin on
  delay(val);                  // stop the program for some time
  digitalWrite(ledPin, LOW);   // turn the ledPin off
  delay(val);                  // stop the program for some time
}
```

Piezo (analog Input):

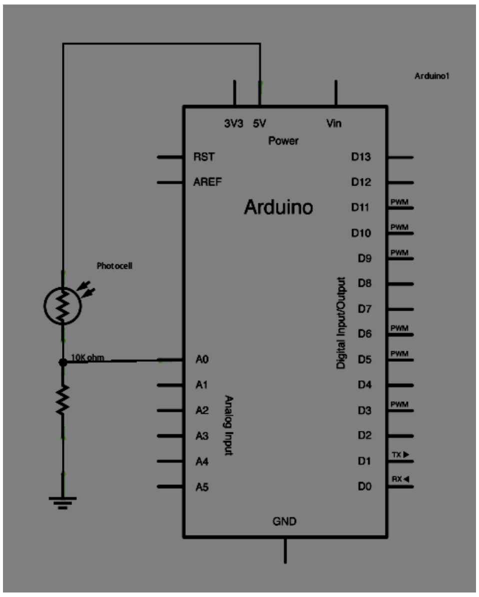
Scematic:

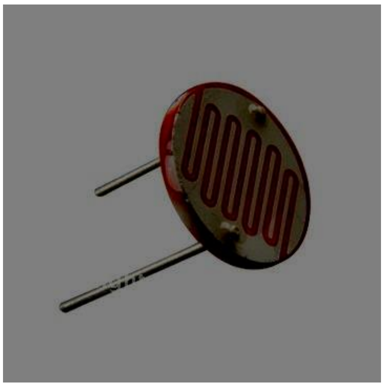
Code:



```
// these constants won't change:
const int ledPin = 13;        // led connected to digital pin 13
const int knockSensor = A0; // the piezo is connected to analog pin 0
const int threshold = 100;  // threshold value to decide when the detected sound is a knock or not

// these variables will change:
int sensorReading = 0;        // variable to store the value read from the sensor pin
int ledState = LOW;           // variable used to store the last LED status, to toggle the light

void setup() {
 pinMode(ledPin, OUTPUT); // declare the ledPin as as OUTPUT
 Serial.begin(9600);        // use the serial port
}
void loop() {
  // read the sensor and store it in the variable sensorReading:
  sensorReading = analogRead(knockSensor);

  // if the sensor reading is greater than the threshold:
  if (sensorReading >= threshold) {
    // toggle the status of the ledPin:
    ledState = !ledState;
    // update the LED pin itself:
    digitalWrite(ledPin, ledState);
    // send the string "Knock!" back to the computer, followed by newline
    Serial.println("Knock!");
  }
  delay(100);  // delay to avoid overloading the serial port buffer }
```

Photo Resistor (analog Input):



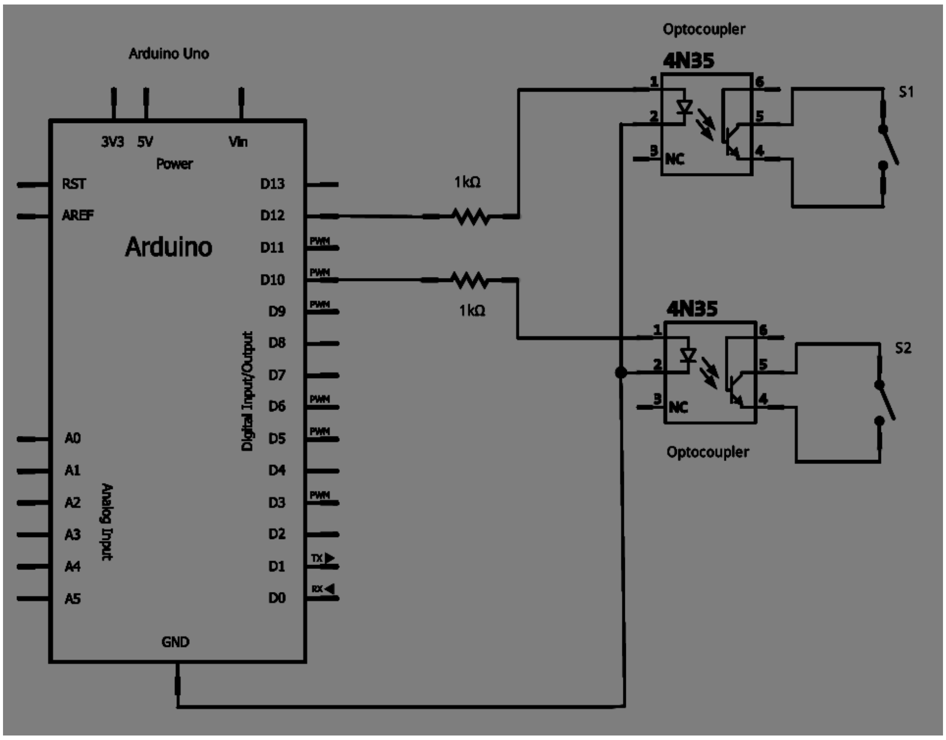


Scematic:
Code:

```
const int sensorMin = 0;        // sensor minimum, discovered through experiment
const int sensorMax = 600;      // sensor maximum, discovered through experiment

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
void loop() {
  // read the sensor:
  int sensorReading = analogRead(A0);
  // map the sensor range to a range of four options:
  int range = map(sensorReading, sensorMin, sensorMax, 0, 3);

  // do something different depending on the
  // range value:
  switch (range) {
  case 0:     // your hand is on the sensor
    Serial.println("dark");
    break;
  case 1:     // your hand is close to the sensor
    Serial.println("dim");
    break;
  case 2:     // your hand is a few inches from the sensor
    Serial.println("medium");
    break;
  case 3:     // your hand is nowhere near the sensor
    Serial.println("bright");
    break;
  }
  delay(1);         // delay in between reads for stability }
```

Optokoppler (digital Output):

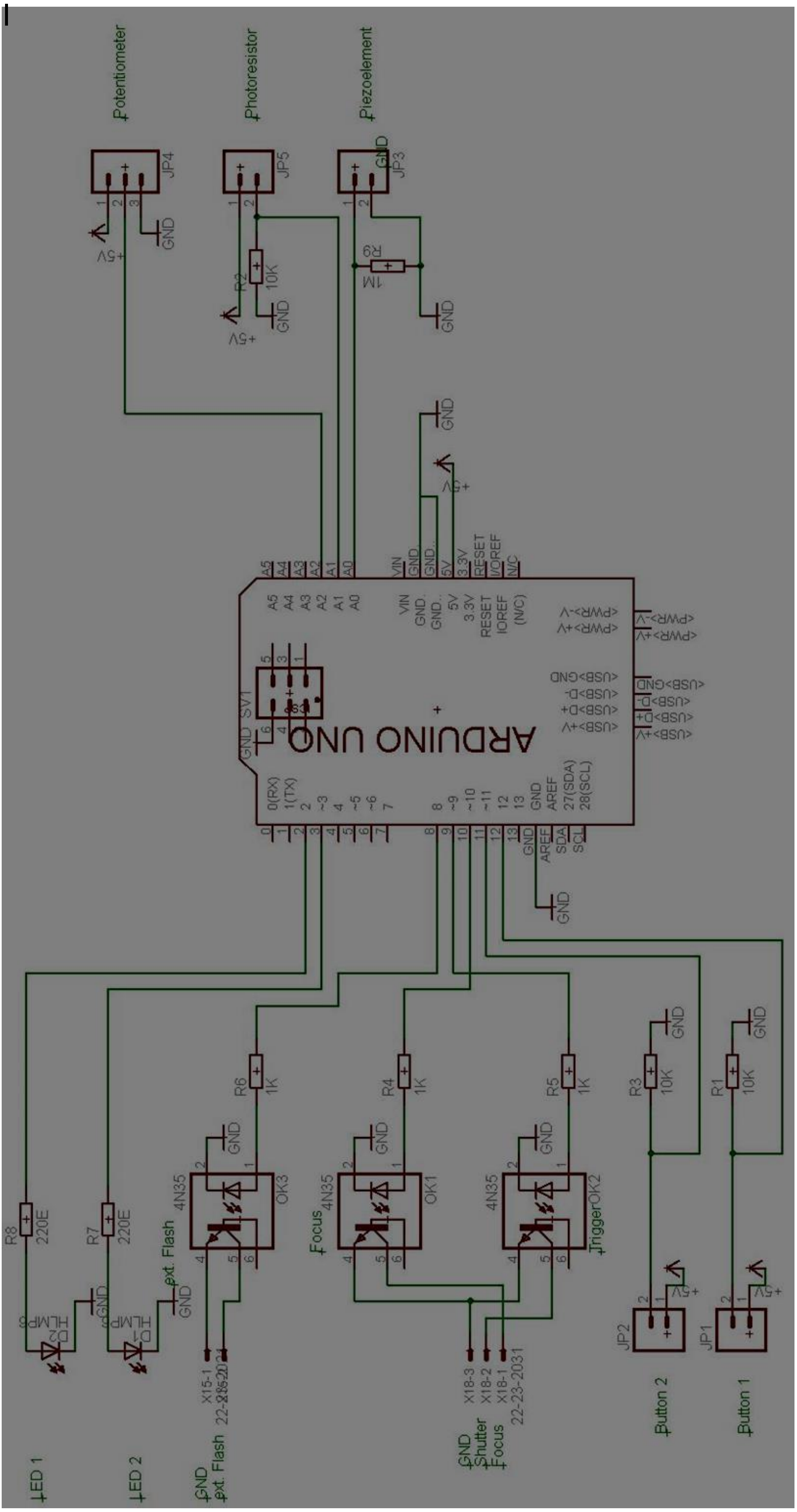
Scematic:



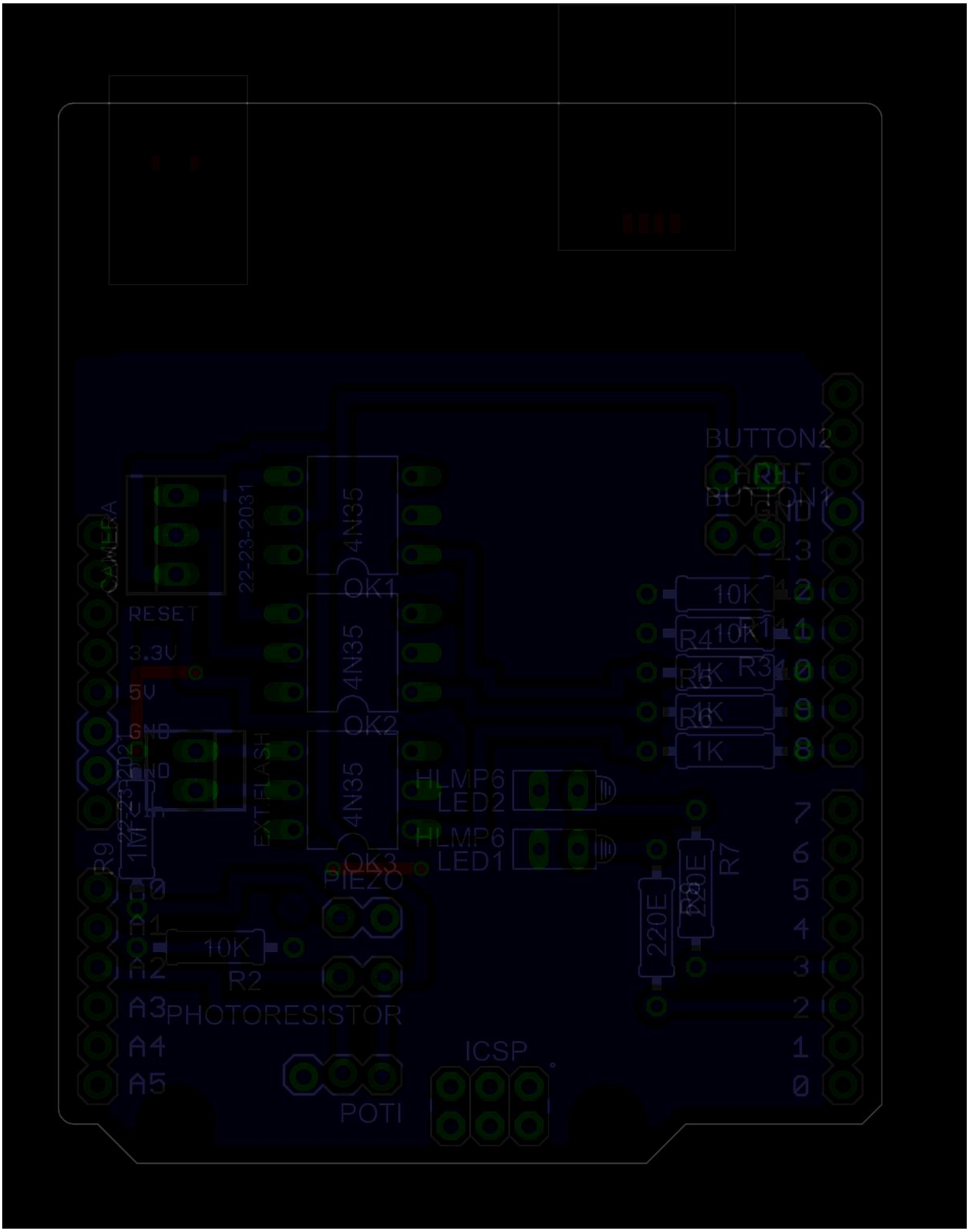
The same testcode as LED – adapt the pinnumbers!!!

Fotomind MINI

Scematic:



Board:



Pinliste:

Analog IN:

A0	IN	Piezo
A1	IN	Photoresistor

A2	IN	Potentiometer

Digital:

D2	OUT	LED1
D3	OUT	LED2
D8	OUT	External Flash
D9	OUT	Camera Trigger
D10	OUT	Camera Focus
D11	IN	BUTTON 1
D12	IN	SWITCH 1

Materialliste:

Wiederstände

1x	1M
3x	10K
3x	1K
2x	220R

Board:

3x	4N35 Optokoppler
3x	Sockellleiste
1x	40Pin Header

1x	Piezo
1x	Photoresistor
1x	Button
1x	Switch
1x	Potentiometer 10K
1x	Potiknopf
2x	LED
2x	Ledhalter

1x	Arduino
1x	Leiterplatte

1x	Stereostecker
1x	Monostecker

1x	Sterenbuchse
2x	Monobuchse

1x	Power Plug
1x	Batteriehalter

Kabel, Lötzinn