

Keychain camera with PIR motion detector controlled by Arduino chip



Keychain cameras are around 10 EUR on Ebay and have relatively good quality and a large storage space on the SD card. Extensive info about these cameras, also named 808 type cameras, is collected here

<http://chucklohr.com/808/C3/index.html>.

I bought two cameras from the same Ebay seller with a month interval and they have different image sensor quality.

With a 8 GB microSD card. 1024x768 pixel photos. 450 kB each. Place for ca 16000 photos. Camera consumes ca 150 mA when ON.

200 mAh LiPo. If each picture is taken during 10 s then energy enough for ca 400 photos from camera LiPo. For taking more pictures need to have an external LiPo.

Applications

PIR activated camera could be used for detecting relatively rare events like animals on the paths in the wood. Could be masked inside a stone, piece of wood or placed high on a tree.

Such cameras are called deer trail cameras

<http://www.golab.com/gct1/gct1.html>

If not placed under a roof then need a waterproof enclosure (food container) or plastic bag. For optical access it would need a glass window (microscope cover glass or antireflection coated glass lens 0 dioptres from optician). PIR plastic cap should be outside, for example glued in a hole using hot glue. Camera could be camouflaged with things growing nearby that are wrapped around with a thread.

As a general rule, do not put camera in private places which no one should see.

If it is a place that anyone can see, like a street or a square then people most probably will not feel upset.

In Germany placing a recording camera at work has to be allowed by authorities as the boss could tell if the workers work. But sometimes it could be useful to see, who has taken tools from workplace in your absence or who has left mess in a kitchen.



Camera could be also put in a car or model airplane or on Helium balloon.

My previous externally triggered camera was Mustek Gsmart that I used to take time lapse photos of a sunrise and put on model airplane.

<http://barbara320.gotdns.com/lnis/elektronika/Gsmart%20camera%20auto%20switch.pdf>

Automatic picture taking for Mustek Gsmart camera was realized before me here

<http://www.rc-cam.com/cammangs.htm>

Arduino chip in low consumption mode

Arduino chip is programmed in Duemillenove board and then removed and put on custom made board.

Before soldering the circuit was tested on a Steckboard (Protoboard) breadboard.

Arduino consumes 10 mA when run at 16 MHz. It is convenient to keep this frequency for serial communication to run and for delays. In power down mode it consumes 0.34 mA. That is fine.

Example: battery powered Arduino chip “Sleep and wakeup by Watchdog” Nightingale sound is here

http://interface.khm.de/index.php/lab/experiments/sleep_watchdog_battery/

Interrupts are described here:

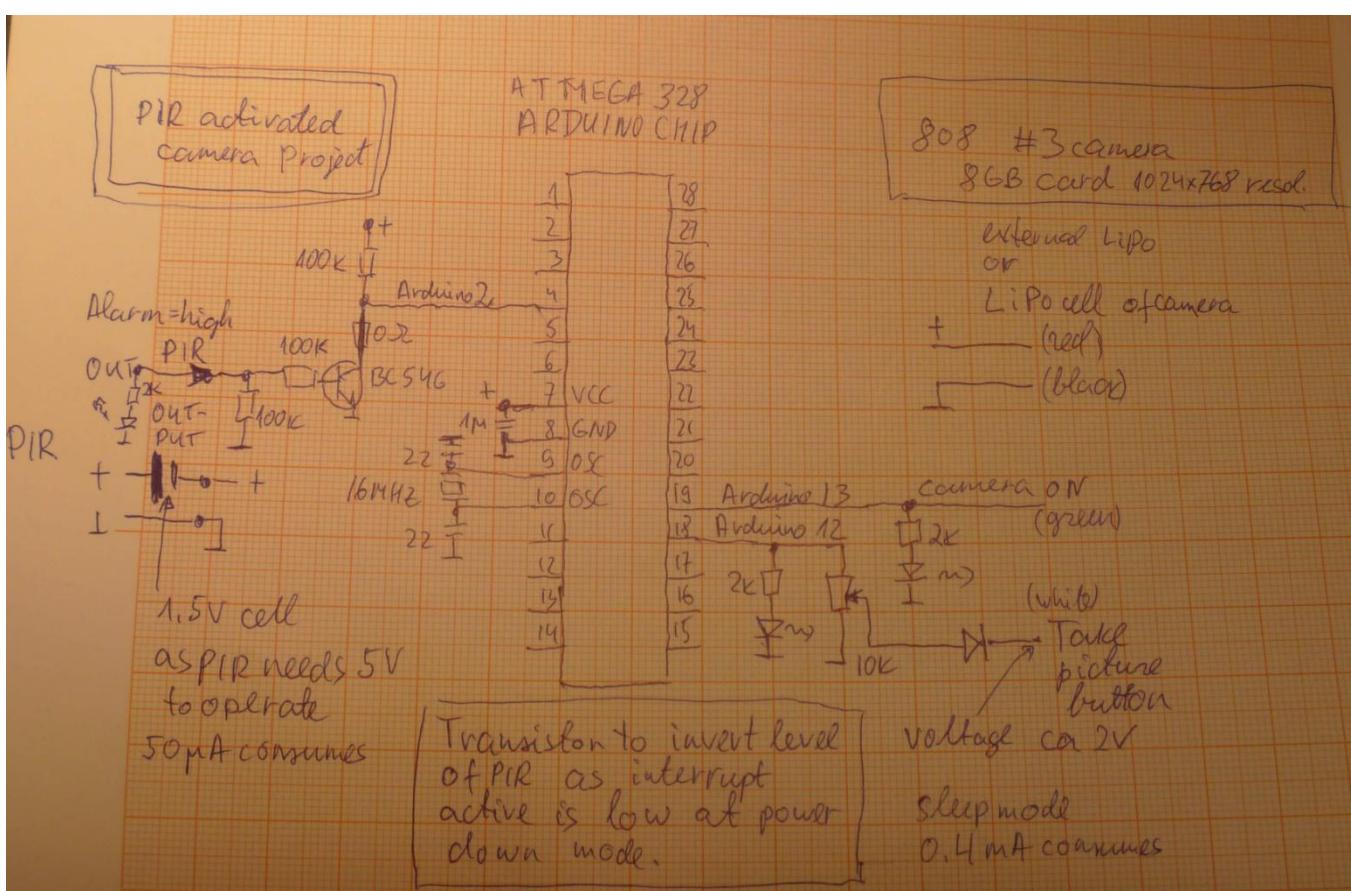
<http://www.arduino.cc/playground/Learning/ArduinoSleepCode>

More on interrupts <http://www.windmeadow.com/node/19>

Actually for Arduino in power down mode the interrupt will happen only when pin goes low. If something else stands in interrupt description it will not go into power down mode.

Motion detector gives High output when activated, so need an inverter. Simple inverter is a transistor. Inspiration to use a transistor is from "Morse Code Alarm Clock Modification"

<http://www.cappels.org/dproj/malc/alarm.html>

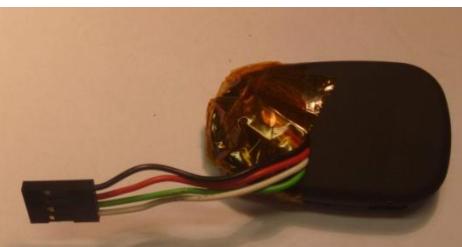
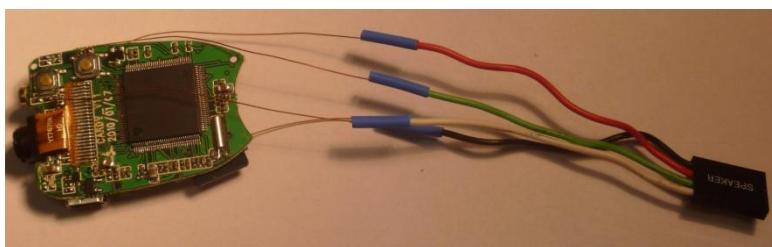
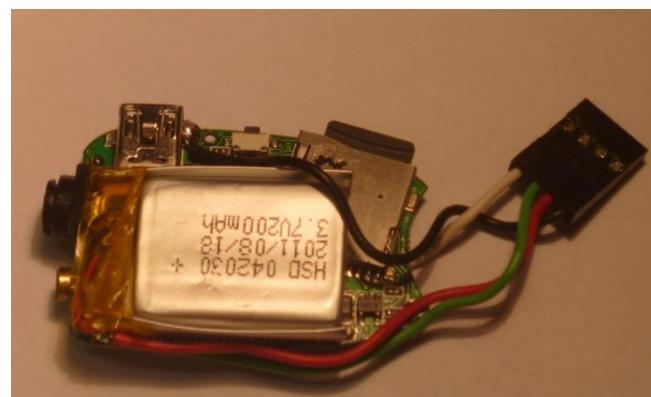
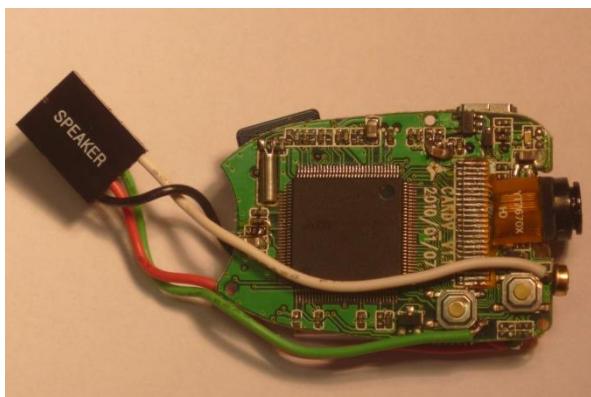


Construction photos

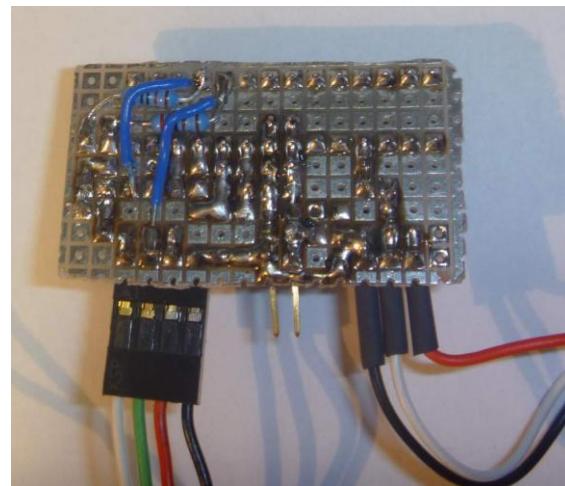
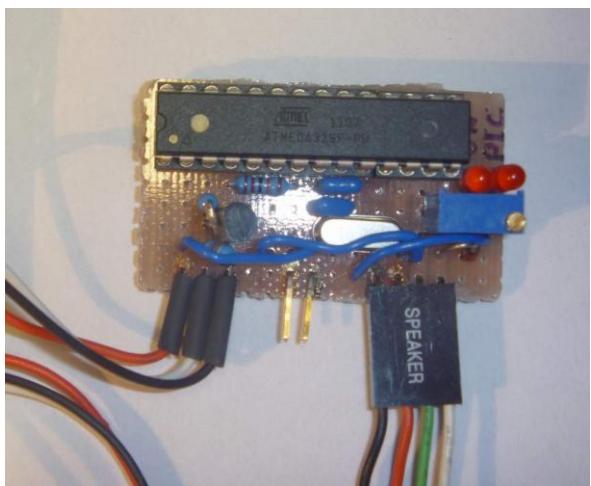
First you can solder connector to buttons and manage to control camera manually. It is better to solder thin wires inside and extend them outside with normal wires. The picture button had to have the voltage level around half of battery voltage. That is why there is extra stuff in schematics. Without a diode camera hang and did not switch off. Anyway, if camera hangs, press the reset button. A correct solution would be to use opto-couplers like in the links below:

<http://www.glacialwanderer.com/hobbyrobotics/?p=10>

<http://diy.viktak.com/2011/03/zeus-lightning-trigger-for-cameras.html>



Pictures showing ATMEGA328 chip removed from Arduino, 16 MHz quartz connectors, LEDs.



PIR motion detector

From Ebay:

5x Adjust Infrared IR PIR Motion Sensor Detector Module US \$12,68 from other.deal

<http://www.ebay.de/itm/5x-Adjust-Infrared-IR-PIR-Motion-Sensor-Detector-Module-/150632915292?trksid=p5197.m7&trkparms=algo%3DLVI%26itu%3DUCI%26otn%3D4%26po%3DLVI%26ps%3D63%26clkid%3D4001763721082665159>



Good sensitivity, large range 7 m. Consumes 60 uA.

Unfortunately it is specified to run from 4.5V and runs unreliable from a single LiPo and needs a 1.5V battery in series with power line to boost voltage.

Another modules

5x Sensor Melder Module PIR Low Voltage Bewegungsmelder

http://www.ebay.de/itm/5-Sensor-Melder-Module-PIR-Low-Voltage-Bewegungsmelder-/270786796651?pt=DE_Haus_Garten_Heimwerker_Sicherheitstechnik&hash=item3f0c269c6b



This is low voltage version that runs from a single LiPo, but range 3 m.

Programming

To take photo

- 1) Switch on by pressing button 2 s.
- 2) Wait 1 s for camera to get running. Led will go on.
- 3) Press picture button. Led will blink 1 time.
- 4) Wait and then switch off by long press off button.

To take video:

- 1) Switch it on. LED will light up.
- 2) Long press button that is closest to front. When LED blinks 3 times video has been started. Led will be off while recording.
- 3) To stop recording, press the start button again and LED should come back on. Video is stopped also when you switch camera off.

Some people report that their cameras need slightly different key combinations.

For correcting time stamp I use program from

<http://www.guncam.de/index.php/de/support>

You need to figure out if your cam is SUN or EKEN version.

Download program and run it while cam is connected via USB. It should set time.

```
// 808 keychain camera control: motion detector wakes up arduino,  
// It switches on camera takes a picture, switches off, sleeps  
#include <avr/sleep.h>  
  
int wakePin = 2;          // pin used for waking up  
int onPin = 13;           // pin used to switch cam on  
int snapPin = 12;         // pin used to take a pic  
  
int sleepStatus = 0;       // variable to store a request for sleep  
int count = 0;             // counter  
  
void wakeUpNow()          // here the interrupt is handled after wakeup  
{  
}  
  
void setup()  
{  
  
    pinMode(wakePin, INPUT);  
    pinMode(onPin, OUTPUT);  
    pinMode(snapPin, OUTPUT);  
    Serial.begin(9600);  
    delay(5000);  
    attachInterrupt(0, wakeUpNow, LOW); // use interrupt 0 (pin 2) and run function  
                                    // wakeUpNow when pin 2 gets LOW  
}  
  
void sleepNow()            // here we put the arduino to sleep  
{
```

```

DDRB = B00000000; // sets Arduino pins as inputs
DDRD = B00000000; // sets Arduino pins as inputs
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); // sleep mode is set here

    sleep_enable(); // enables the sleep bit in the mcucr register
                    // so sleep is possible. just a safety pin

attachInterrupt(0,wakeUpNow, LOW); // use interrupt 0 (pin 2) and run function
                                // wakeUpNow when pin 2 gets LOW

sleep_mode(); // here the device is actually put to sleep!
// THE PROGRAM CONTINUES FROM HERE AFTER WAKING UP

sleep_disable(); // first thing after waking from sleep:
                  // disable sleep...
detachInterrupt(0); // disables interrupt 0 on pin 2 so the
                     // wakeUpNow code will not be executed
                     // during normal running time.

pinMode(onPin, OUTPUT);
pinMode(snapPin, OUTPUT);

}

void loop()
{
    // display information about the counter
    Serial.println("Awake");
    // if (pir == 1)
    Serial.println("Switch On");
    digitalWrite(onPin, 1);
    delay(2000);
    digitalWrite(onPin, 0);
    delay(1000);

    Serial.println("Take pic");
    digitalWrite(snapPin, 1);
    delay(500);
    digitalWrite(snapPin, 0);
    delay(2000);

    Serial.println("Switch off");
    digitalWrite(onPin, 1);
    delay(3000);
    digitalWrite(onPin, 0);
    delay(8000);

    Serial.println("Entering Sleep mode");
    delay(100); // this delay is needed, the sleep
                //function will provoke a Serial error otherwise!
    sleepNow();
}

```