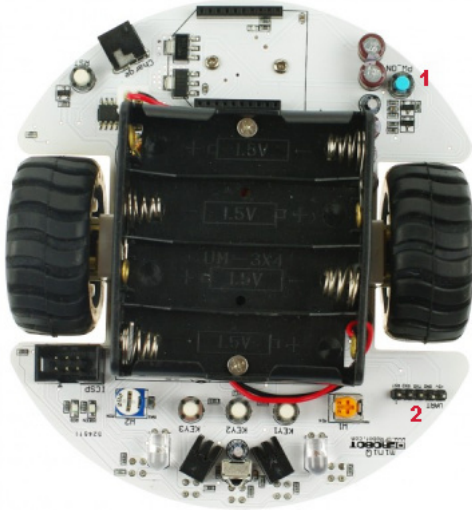


## Robot programming with Arduino (Part 1)

Robotics is one of the most fascinating technical areas. But so far, it was almost exclusively reserved for absolute experts, because the software development was extreme and the available hardware prohibitive. But now there are cheap entry-level models and highly efficient software technologies that enable everyone, this mysterious technology is needed, because once you get it admits, to get rid of it again difficult o) The possibilities that are opening up fascinating and inexhaustible at the same time.

But if you dare, then let us take the first steps together. We choose as the hardware [DFRobot ROB0081](#) and for software development, the Arduino system. If you have never look first of all our tutorial " [What is Arduino or DFRduino](#) "to. There you will learn the fundamentals.

The communication with the robot



After you have inserted the batteries into the holder and the blue button "PW\_ON" (No. 1 in picture) have pressed lights four blue LEDs under the robot and give a first sign o controller contains quite reaching its bootloader no data yet that is all what he gives of himself.

To him breathe a little more life, we must first make the programming adapter Xbee USB ready. The small board is the connection between USB interface and robot and use Labs. The driver for this board, please contact:

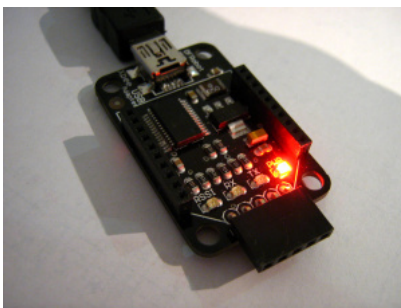
CP210 drivers for Windows

CP210 drivers for Linux

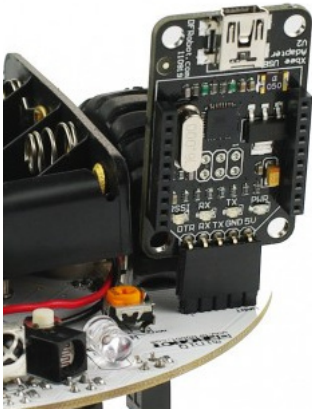
CP210 drivers for MacOSX

The installation is very easy. You run the program and then a new serial port located on your PC, you must specify for communication. But we later come.

Now connect the PC and the XBee board via the USB cable.



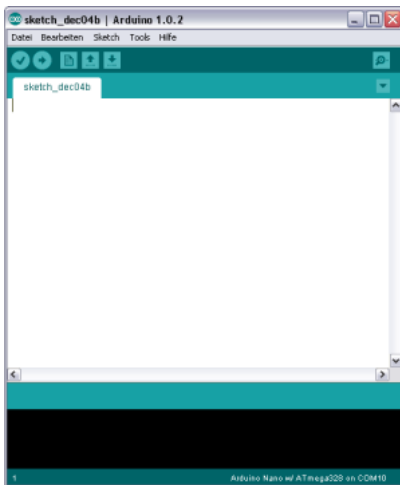
Then plug the Xbee USB adapter on the connector (No. 2 in the figure) of the board of the robot.



The first program

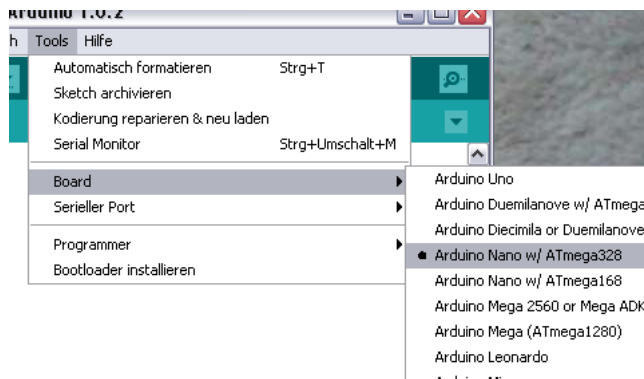
The connection from the computer to the robot has now been established and we can create a first small program to verify that all components are working as desired. We use a little buzzer.

We work on the Arduino IDE. The program requires no installation routine and is not registered with Windows in the registry. Just unzip and open the arduino.exe. You should see the following picture:



With the Arduino development environment, we can create programs in C, compile and transmit to the hardware. But for this you must first make another 2 settings. First, select the serial interface over which the Xbee USB board based communication. We do this via a click to the menu "Tools -> Serial Port". The relevant interface is usually associated with COM10. If you are not sure, look in Control Panel after (see tutorial "[What is Arduino or DFRduino](#)").

The second adjustment relates to the controller we want to program. In our case, this is the "Arduino Nano ATmega328".



The configuration is already done and we can finally create the first program.

To do this, enter the following code:

```

#define BUZZER 11
void setup()
{
    pinMode(BUZZER,OUTPUT);
}
void loop()
{
    unsigned char i,j;
    while(1)
    {

        for(i=0;i<80;i++)
        {
            digitalWrite(BUZZER,HIGH);
            delay(1);
            digitalWrite(BUZZER,LOW);
            delay(1);
        }

        for(i=0;i<100;i++)
        {
            digitalWrite(BUZZER,HIGH);
            delay(2);
            digitalWrite(BUZZER,LOW);
            delay(2);
        }
    }
}

```

The first line defines the symbolic expression BUZZER the value 11 That is wherever the compiler on the word BUZZER encounters, it is replaced by the number 11. 11 is not "Digital Pin 11", so the connector 11 which is connected to the buzzer (buzzer). About the compiler command (directive) # define, you can assign a symbolic expression me

An Arduino program (which is now referred to as sketch), there is usually a setup-part by the connections are defined, and a loop-part, ie a loop in which the actual program also has our mini program.

The "pinMode ()" function is part of the Arduino library, and allows you to configure the controller ports. With the command " pinMode (BUZZER OUTPUT) "we switch the cor in the starting mode.

In the following loop comes another Arduino function to use: "digitalWrite ()". The first Parameters of this function is the port that the second determines the level. The third A use is "delay ()", a wait function that returns a value as a parameter in milliseconds. Now let's look at the whole thing in the end:

The for loop is executed 80 times. Here, the pin 11 is first for 1 millisecond is set to "high", and then for 1 millisecond to "low". Which corresponds to a period of 2 ms, that is with a frequency of 500Hz, followed by a loop that is executed 100 times, the delay is now 2 ms, the period of 4 ms, which in turn at a frequency of 250 Hz corresponds.

Thus, our first program would be ready. For the next two steps "Check / Compile" and "Transfer" click once on the hook left in the menu bar and then click the arrow next to i



The program is now on the ROB0081 and you should hear the results if you switch the robot on the blue button. The first Step is done and if you want more, please check be forward to meeting you!

**For more information and orders of the  
ROB0081 MiniQ robot 2WD  
simply click on the following link: [DFRobot ROB0081](#)  
**NEW** experimental robot [RoverRom- B3000](#) \_**

They want to be Arduino professional? Then our [video microcontroller course MC1](#) just the thing for you!

**If you have questions about, please contact us.  
We are happy to help!**

**Your team of Böcker system electronics**

Note: Our Tutorial offer will be extended at irregular intervals. If you want to be informed immediately of any new publication, please sign up [here](#) in our notification list "Tut unsubscribe in your account or by e-mail this courtesy at all times.

Copyright © Böcker system electronics