

# GeoSensor Short Manual

## Building a transducer device

*This manual is intended for persons having basic knowledge of electronic circuits and Arduino programming. If you are unsure what some of the instructions in this manual mean, please read some introduction to the Arduino platform first.*

Materials needed:

- An Arduino board
- A USB cable for the Arduino board
- A Bluetooth serial pass-through module (recommended: HC-05)
- Jumper wire
- A battery or other power source
- The sensors that should be read out
- Three LEDs
- Three resistors (around 1 kOhm, used for the LEDs)
- One push-button
- One breadboard
- If desired: a GPS module

Steps to build the transducer:

- 1) Wire the regulated voltage output of the Arduino Board and the ground to the breadboard. Use the voltage level the Arduino electronics use.
- 2) Wire the push button between the Arduino pin 2 and ground
- 3) Wire two of the LEDs between arbitrary Arduino digital pins (positive side of the LED) and a resistor connected towards ground.
- 4) Connect the Bluetooth module to the RX and TX pins of the Arduino. The pins must be crossed, Arduino RX must be connected to Bluetooth TX and vice-versa. If your Bluetooth module and your Arduino use different voltages, you must include a resistor voltage divider or some other protection circuitry.
- 5) Connect the power lines of the Bluetooth module.
- 6) If available, connect the positive side of the last LED to the Bluetooth modules state pin. The negative side should be connected to a resistor connected to ground.
- 7) Connect the sensors as described in the sensors documentation.
- 8) Read the code comments in the GeoSensor software template. You only need to read until the `getComment()` function, the rest should be left unchanged.

- 9) Change all the constants in the code to suit your model. Even if you don't use GPS you can connect the GPS LED as a status Indicator to see the system is on.
- 10) If you want to connect a GPS module, connect it to another serial port if available or to two arbitrary digital pins. Configure the software accordingly.
- 11) Get the libraries and the code needed to read out your sensors ready.
- 12) Include the code to read out the sensors in the `getSensorJson()`-function. You can include as many sensors as the SRAM of the Arduino allows you to, but the data fields must be formatted exactly as in the example given in the comment above the function.
- 13) Flash the software to the Arduino board.
- 14) If the system does not work, set the Bluetooth serial to Serial and read the output using the Arduino serial monitor.

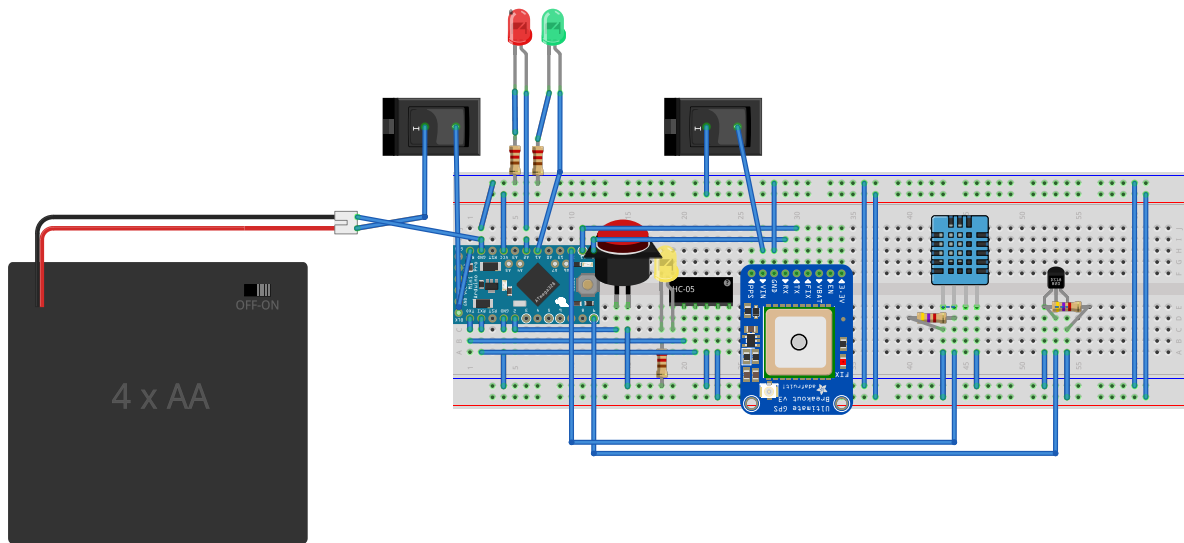
#### Troubleshooting:

- If there is no output when the button is pressed, either the Button is not connected properly or the configuration is not OK. The button needs to be connected to a pin that supports interrupts. Read the Arduinos datasheet.
- Most problems are caused by an inappropriate JSON buffer size. If the message sent does not contain the information it should, try adjusting the buffer size or reducing the amount of information (test turning the GPS off).
- The output data format is JSON. If the data looks strange, try to understand the underlying structure.
- The Baud rate used in the communication to the Bluetooth module must be set correctly.

#### Note:

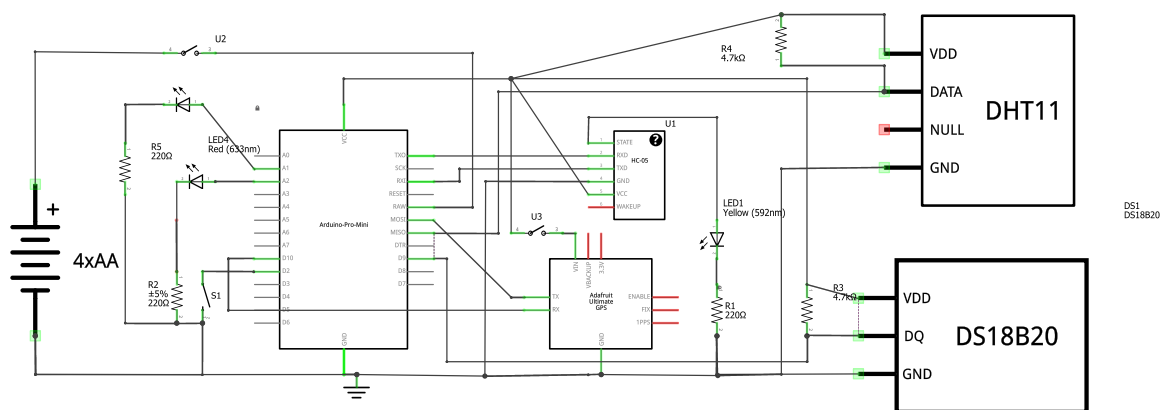
If the same serial port is used for programming and the Bluetooth communications, only one of the connections should be connected at any time.

**Example for a DHT11 and a DS18B20 sensor and an Adafruit GPS module connected to an Arduino Pro Mini 3.3 V:**



fritzing

DHT11



fritzing