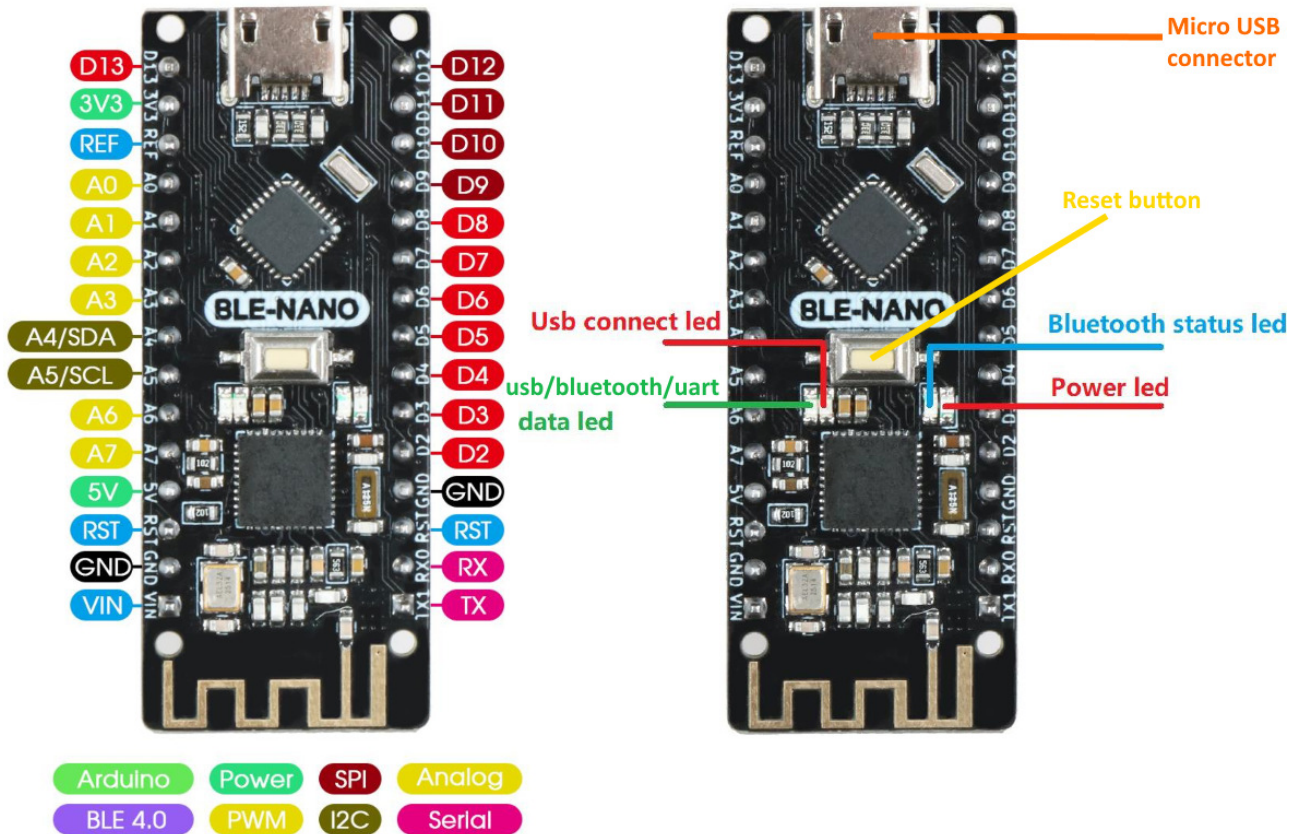


...ntroduction to the Arduino Nano BLE

🕒 April 10, 2020 📁 Education 🔗 arduino, ble, bluetooth, communication, nano, nano-ble

The Nano BLE as sold [here](#) is a combination of a traditional Nano with a Bluetooth BLE module, based on the CC2540 BLE bluetooth chip. The Bluetooth module is directly [connected to the TX, RX pins of the Arduino](#) (pins 0 and 1).

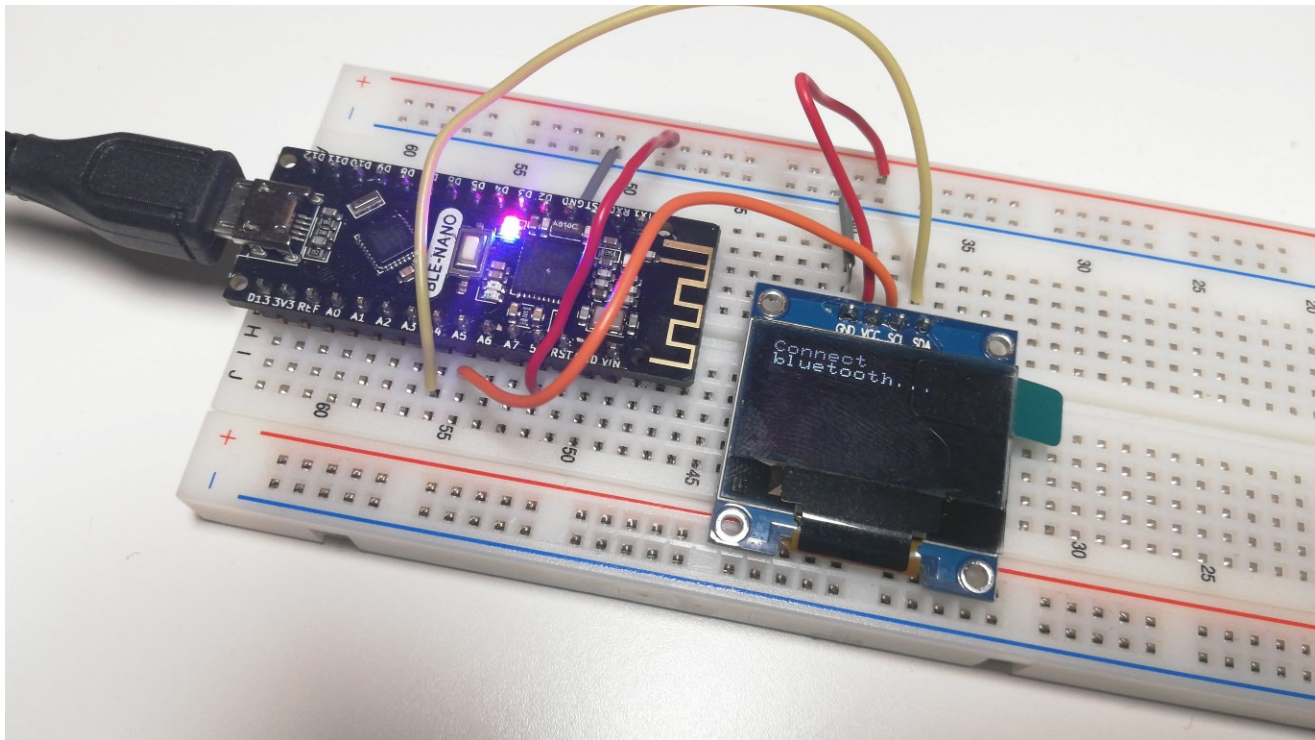
This is the pinout of the board:



Check out the location of the 4 status leds on the board and the reset button.

A first try with Bluetooth

First, add a display to the Arduino, eg. by [following this tutorial](#) "Using an OLED Display with Arduino":



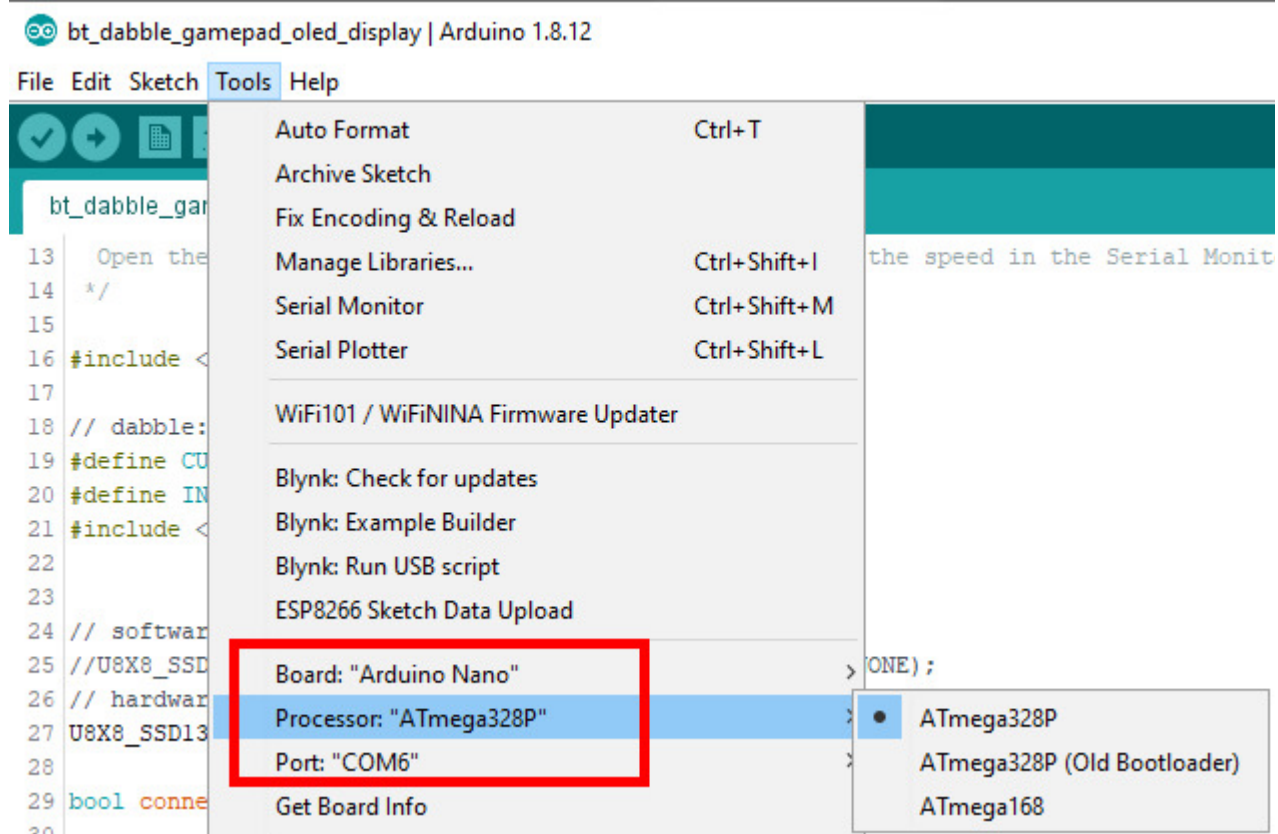
Install Dabble on your Smartphone

We will use the [Dabble App](#) to communicate with the Nano-BLE. Dabble is available for [Android](#) and [iOS](#).


Upload a test sketch

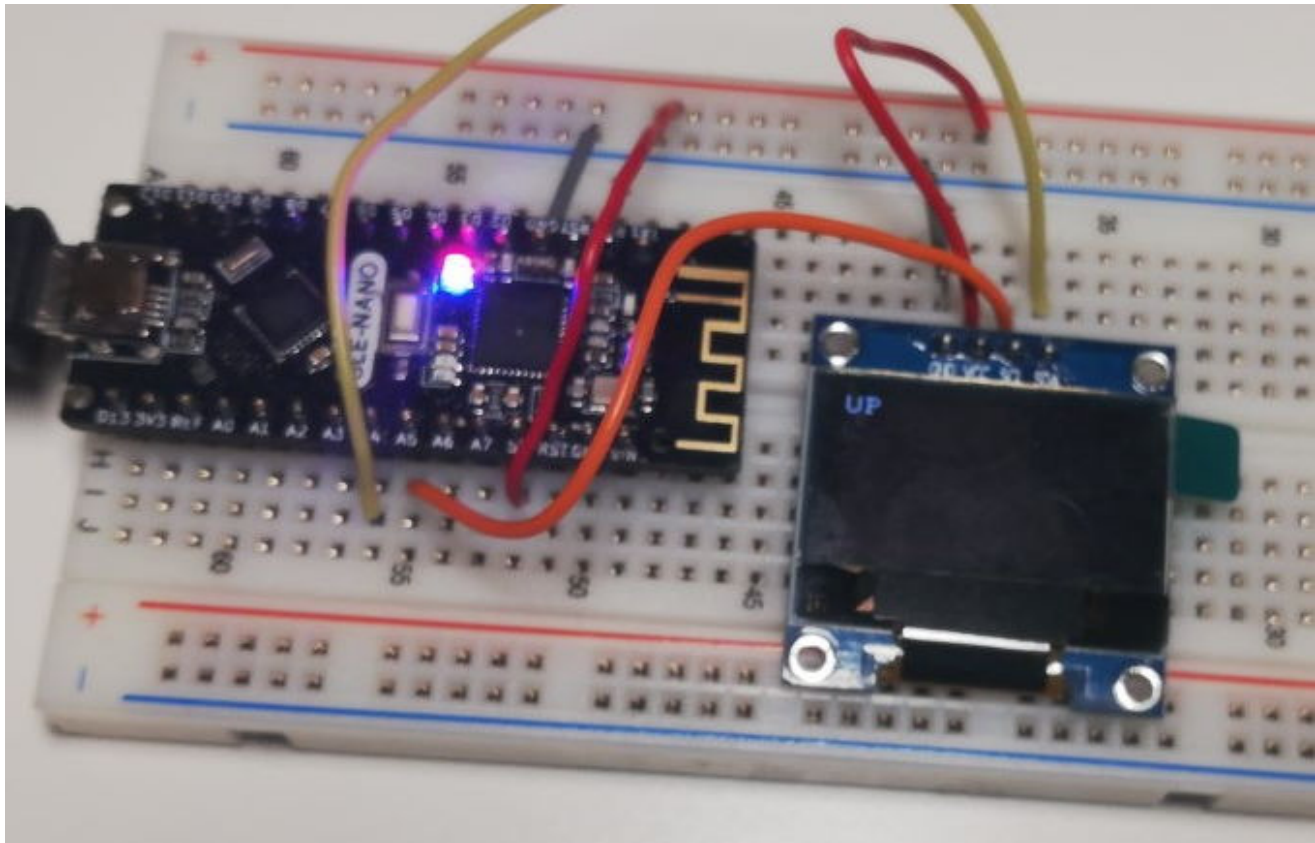
Download this sketch [bt_dabble_gamepad_oled_display.ino](#) and open it in the Arduino IDE. This sketch uses the libraries “Dabble” and “U8g2” so check if these are already installed (*Tools > Manage Libraries*). If not, install them. ([Learn more about Arduino libraries](#))

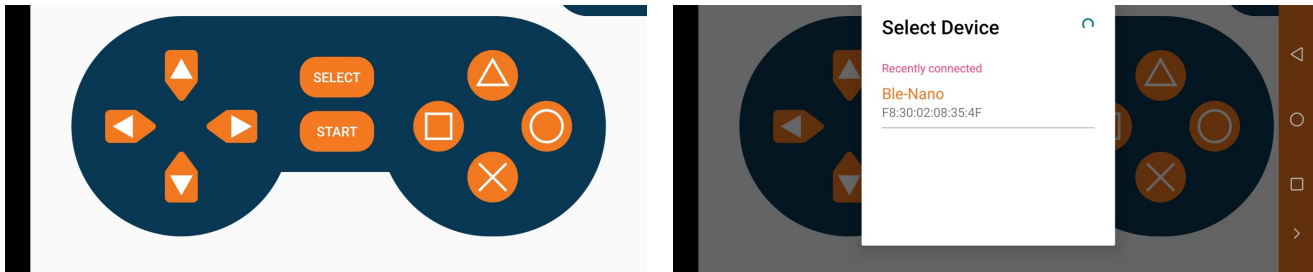
Connect the Arduino to your computer with a micro USB cable. In the Arduino IDE set the board details (your Port is probably different!):



Next, upload  the sketch.

After the upload is successful, on your phone turn on Bluetooth. Start the Dabble app and connect  to the Arduino (default Bluetooth name: "Ble-Nano"). Select the Gamepad tile. You should now see the button pressed on the display connected to the Arduino:





Bluetooth communication between two Nano BLE devices

This is an advanced topic. Make sure you have experience in Arduino programming and [learn the basics of Bluetooth first](#). For an overview of AT-commands for the Nano BLE, [read this reference](#). It contains a list of all AT-commands available for the Nano BLE.

In the Bluetooth communication protocol, the master initiates the connection. A master can connect to up to 7 slaves. Once connected both master and slave can interchange information without restriction (both ways).

The Bluetooth module of the Nano BLE communicates via the Serial connection of the Arduino. This has some advantages, and some disadvantages. Advantage is that no pins are occupied for the Bluetooth module. Disadvantage is that communication may sometimes interfere with regular communication (eg. while uploading or communicating via the Serial Monitor).

Make sure you read these notes and understand them before you continue:

- Some AT-commands will restart the Arduino, which causes the Serial Monitor to not respond to further AT-commands. In that case, restart the Serial Monitor. If that also does not work, close the Serial Monitor, disconnect the USB-cable, then connect it again and start the Serial Monitor.
- If a sketch is running which processes Serial data (eg. using `Serial.read()`), you cannot send AT-Commands via the Serial Monitor. If necessary, upload an empty sketch before sending AT-Commands.
- If a Bluetooth connection is active, the device cannot receive AT-commands and also not upload new sketches. Disconnect first (eg. by powering off one of the devices).
- If a Bluetooth connection is established, the Bluetooth status LED is on. If there is no connection, it blinks.

Get device address of slave:

Since the Nano BLE by default is a slave, we do not have to configure anything. But we must get the device address to be able to connect to it.

Start the Arduino IDE. Make a new sketch. Save it as “empty”.

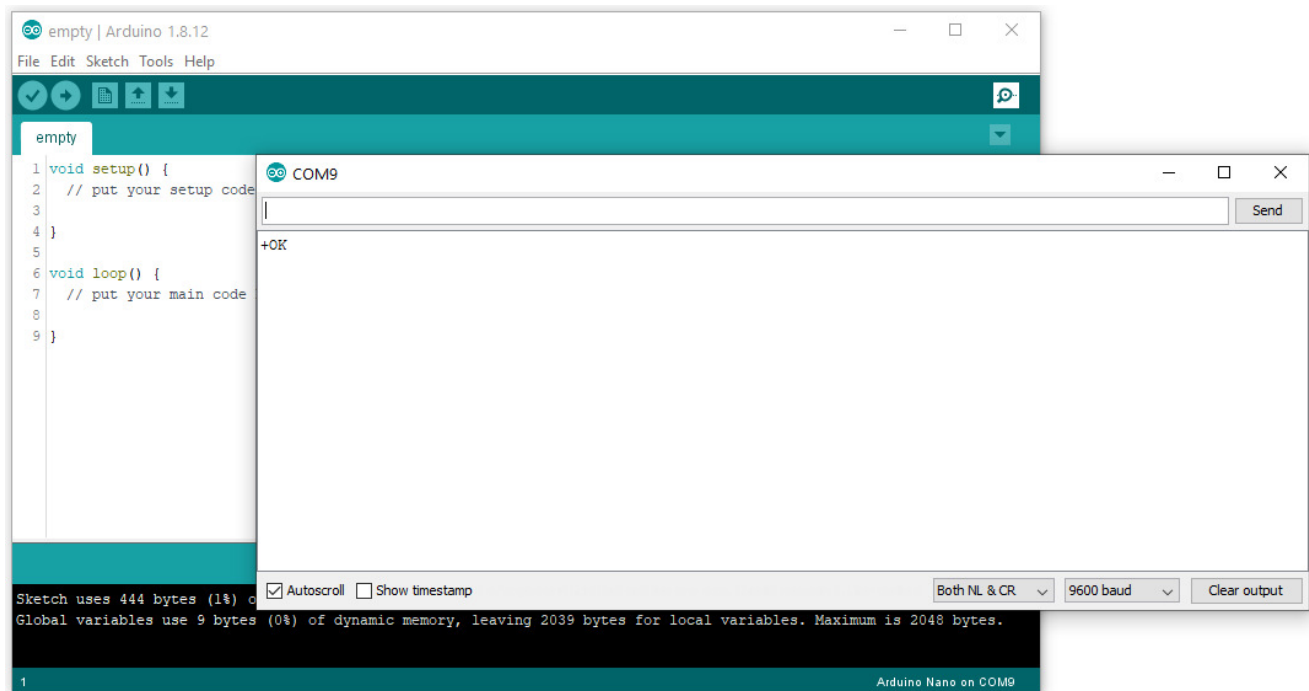
Connect the slave to a USB-port of your computer. Set the proper Board, Processor and Port.

Upload the empty sketch.

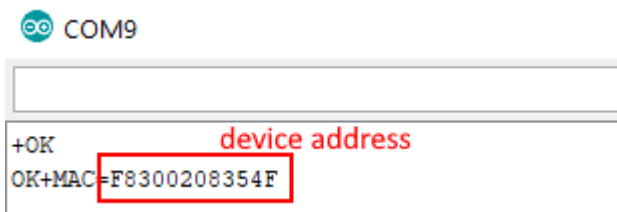
Start the Serial Monitor, to issue AT-Commands in the text field on top.

First, send: `AT` (press Enter)

If you receive an “+OK”, sending commands works:



Now send: AT+MAC to query the device address and write down the device address (eg. copy, paste into Notepad).



Setup master:

Connect the master to a USB-port of your computer.

Connect the slave to a power source (eg. a usb-charger), but not to the computer (to avoid mix-up with the master).

Start the Serial Monitor, to issue AT-Commands.

First, send: AT

If you receive an "+OK", sending commands works.

If it does not work, upload an empty sketch to the Arduino, as the sketch might prevent communication.

Then try again to send AT.

Next, query the role: AT+ROLE it should be 1 (slave) by default.

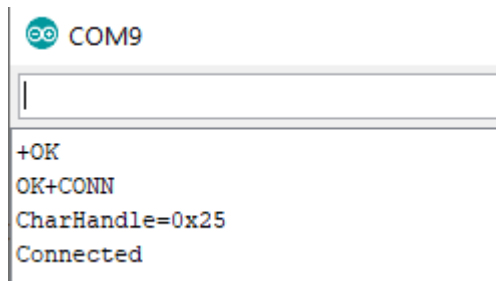
If it is a 1 (slave), set it to master using: AT+ROLE=0

Then query the role again: AT+ROLE it should be 0 (RoleMode=0, master) now.

Do a scan: AT+SCAN

It should return a list of device addresses of Bluetooth devices nearby (this can take a while). The slave should be among them.

Now connect to the Slave: AT+CON=F8300208354F (replace address with the address of your slave!). You will see a message "Connected" and the Bluetooth status led on both Arduino's is on.

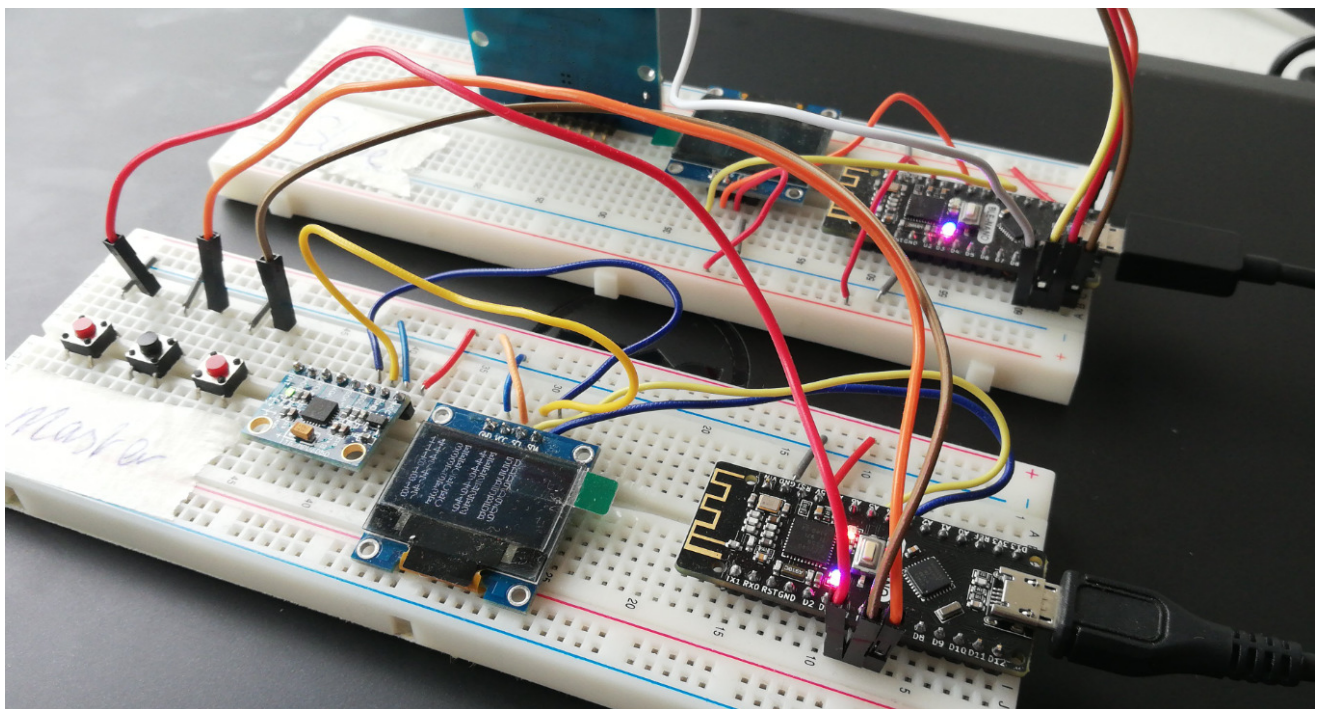


You can also use AT-commands in the Arduino sketch. Eg. in the sketch for the master, to initiate a connection to the slave when the master starts (the slave must be already on!):

```
void setup() {  
    // Open serial communication  
    Serial.begin(9600);  
    delay(2000); // wait 2 sec  
    Serial.println("AT+CON=F8300208354F"); // connect to slave with given  
    address  
}
```

As a helper, you can use this simple example sketch `bt_oled_display.ino` to display incoming data on the display, eg. on the master, to receive data from a slave with a sensor or reader:

As an example, you could [build a RFID reader](#) (as a slave), with the example sketch `rfidscan_display.ino`, and make it send the scanned id (UID) of a card or tag and send it via Bluetooth to the connected device (master):



More information

- [Keywish nano plus](#), Github site with documentation and test app from the manufacturer.

- A similar Nano BLE: [DFRobot Bluno Nano](#)