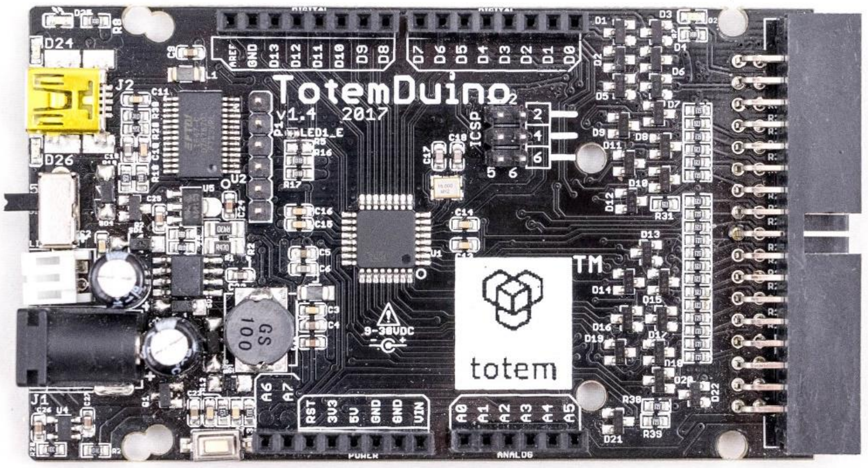
totem

TotemDuino User Guide

www.totemmaker.net



Totem by UAB Aldrea

ver 1.1

Table of Contents

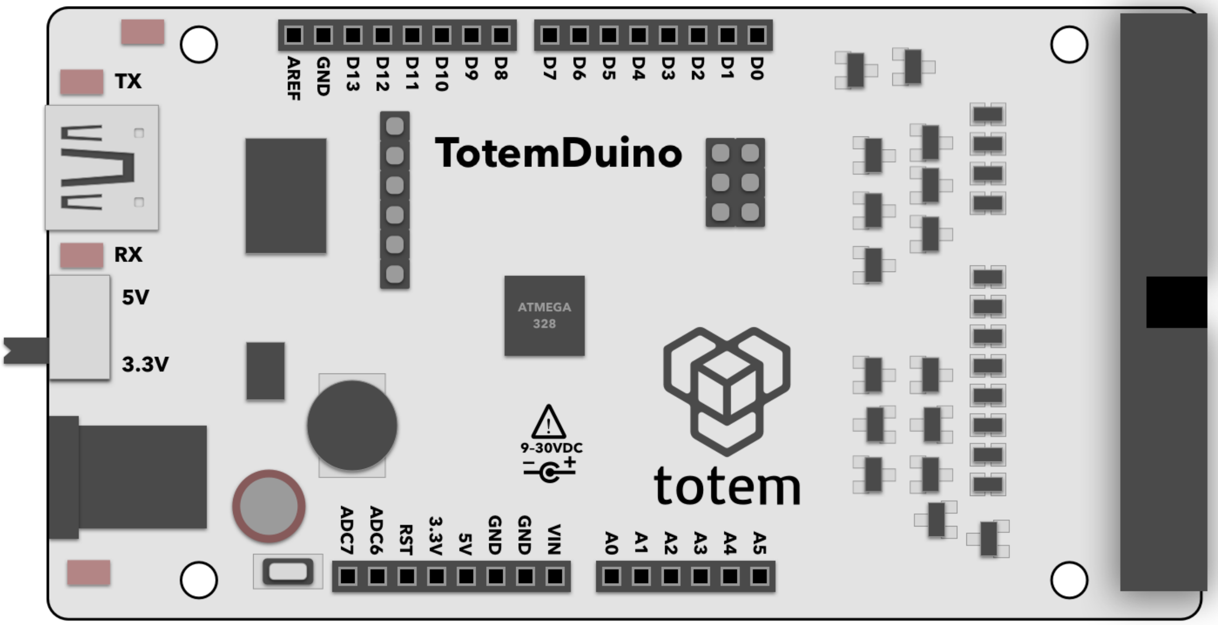


Fig. 1 TotemDuino board

Features

- Atmega328 microcontroller
- 100% Arduino Uno compatible, works with all Arduino shields
- Works in Arduino IDE
- 9-30 V DC or 5V USB supply power range
- When powered from DC power adapter, can supply up to 1.5A of current at 5V.
- Selectable 5V or 3.3V logic voltage
- Noise immune design
- Additional filtering for precise analog voltage measurements
- Integrated programmer - no extra parts needed to start coding.
- Arduino and Totem compatible mounting holes
- 34-pin flat cable connector for integration with Totem MiniLab

Getting started

Starting to use TotemDuino is really simple —it is a fully backwards compatible with Arduino UNO platform, so all the great Arduino IDE features can be used to write firmware for TotemDuino as well.

Connecting TotemDuino to your computer

Using a mini USB cable you can upload new firmware sketches into TotemDuino. It holds a complete backwards software compatibility with Arduino UNO boards, so Arduino IDE can be used to write programs for it. While you can use different programming environments to write firmware for it, using Arduino is one of the most friendliest and quickest way to start.

Firstly, install Arduino IDE to your computer — refer to official documentation for the Arduino IDE installation instructions, which can be found here:

https://arduino.cc/en/Guide/HomePage

Additional learning and introductory resources is recommended to use, such as:

- **What is an Arduino?**
https://learn.sparkfun.com/tutorials/what-is-an-arduino

- **What is a circuit?**
http://learn.sparkfun.com/tutorials/what-is-a-circuit

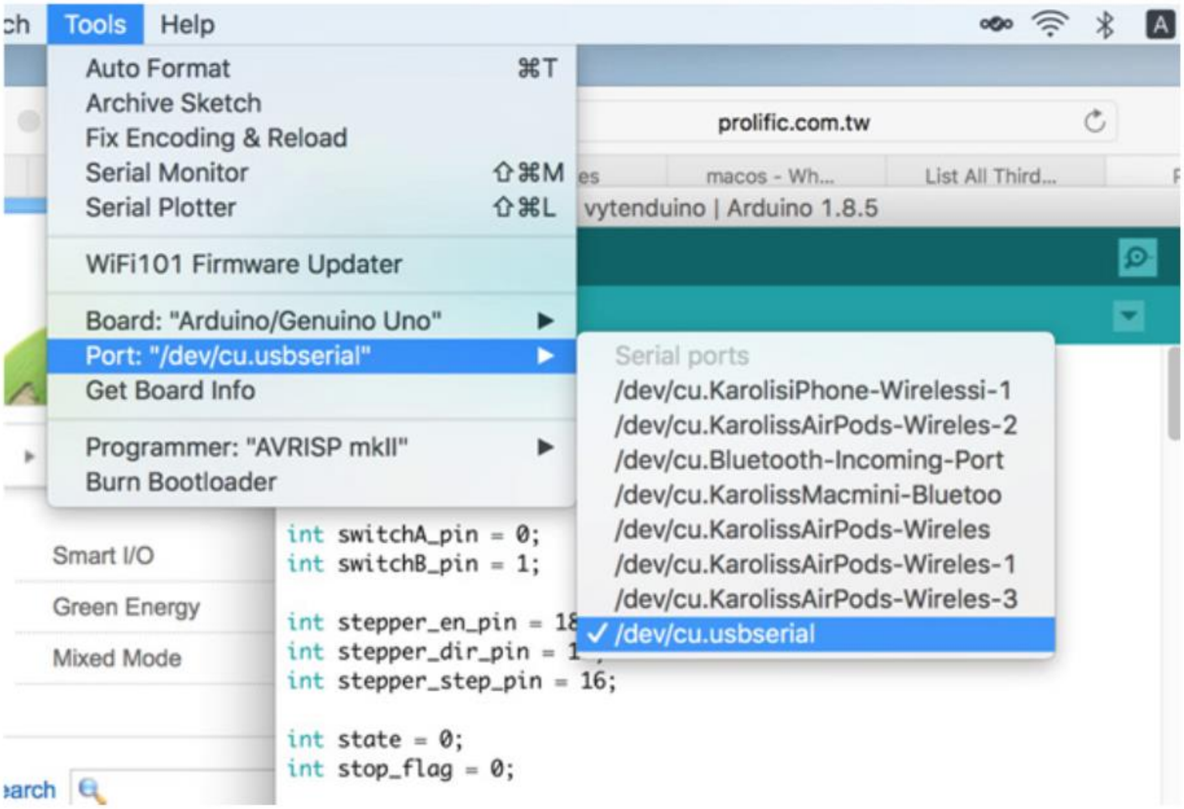
While TotemDuino works without any additional drivers with Windows10, previous versions of Windows might require manually downloading drivers for a communication chip used in it (PL-2303HXD). A latest version can be found at manufacturers website:

http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41

On macOS systems, driver will need to be manually installed for it to work:

http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=229&pcid=41

After installation, TotemDuino serial port **/dev/cu.usbserial** should show up when connected:


ch
Tools
Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno" ▶
Port: "/dev/cu.usbserial" ▶
Get Board Info
Programmer: "AVRISP mkII" ▶
Burn Bootloader
prolific.com.tw
es
macos - Wh...
List All Third...
vytenduino | Arduino 1.8.5
Serial ports
/dev/cu.KarolisiPhone-Wirelessi-1
/dev/cu.KarolissAirPods-Wireles-2
/dev/cu.Bluetooth-Incoming-Port
/dev/cu.KarolissMacmini-Bluetoo
/dev/cu.KarolissAirPods-Wireles
/dev/cu.KarolissAirPods-Wireles-1
/dev/cu.KarolissAirPods-Wireles-3
✓ /dev/cu.usbserial
Smart I/O
Green Energy
Mixed Mode
earch

```
int switchA_pin = 0;
int switchB_pin = 1;

int stepper_en_pin = 18
int stepper_dir_pin = 1
int stepper_step_pin = 16;

int state = 0;
int stop_flag = 0;
```

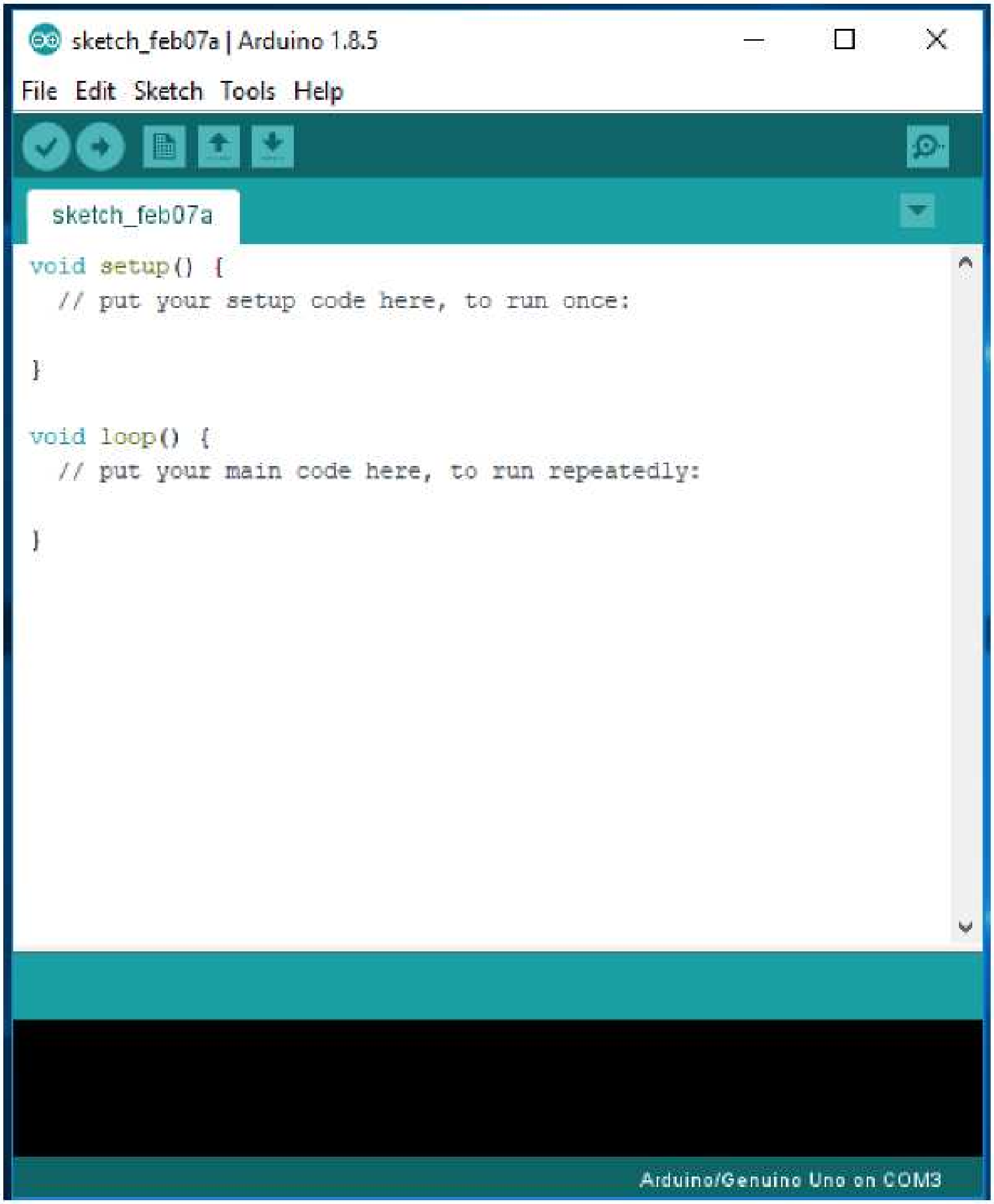
Fig. 2 Selecting TotemDuino board port

One note regarding macOS compatibility is that it’s important to not unplug the device while it’s being programmed. Doing so will cause the driver to get stuck, and further connected TotemDuino boards will not show up until computer is restarted.

On most Linux systems driver for the TotemDuino should be included in the system already, so no further steps are needed to be taken. If it’s missing, or the device doesn’t enumerate when connected, refer to your distribution manual on how to enable PL2303 kernel module. Tested distributions that include the driver are Debian (and derivatives such as Ubuntu) and Fedora.

Using Arduino IDE

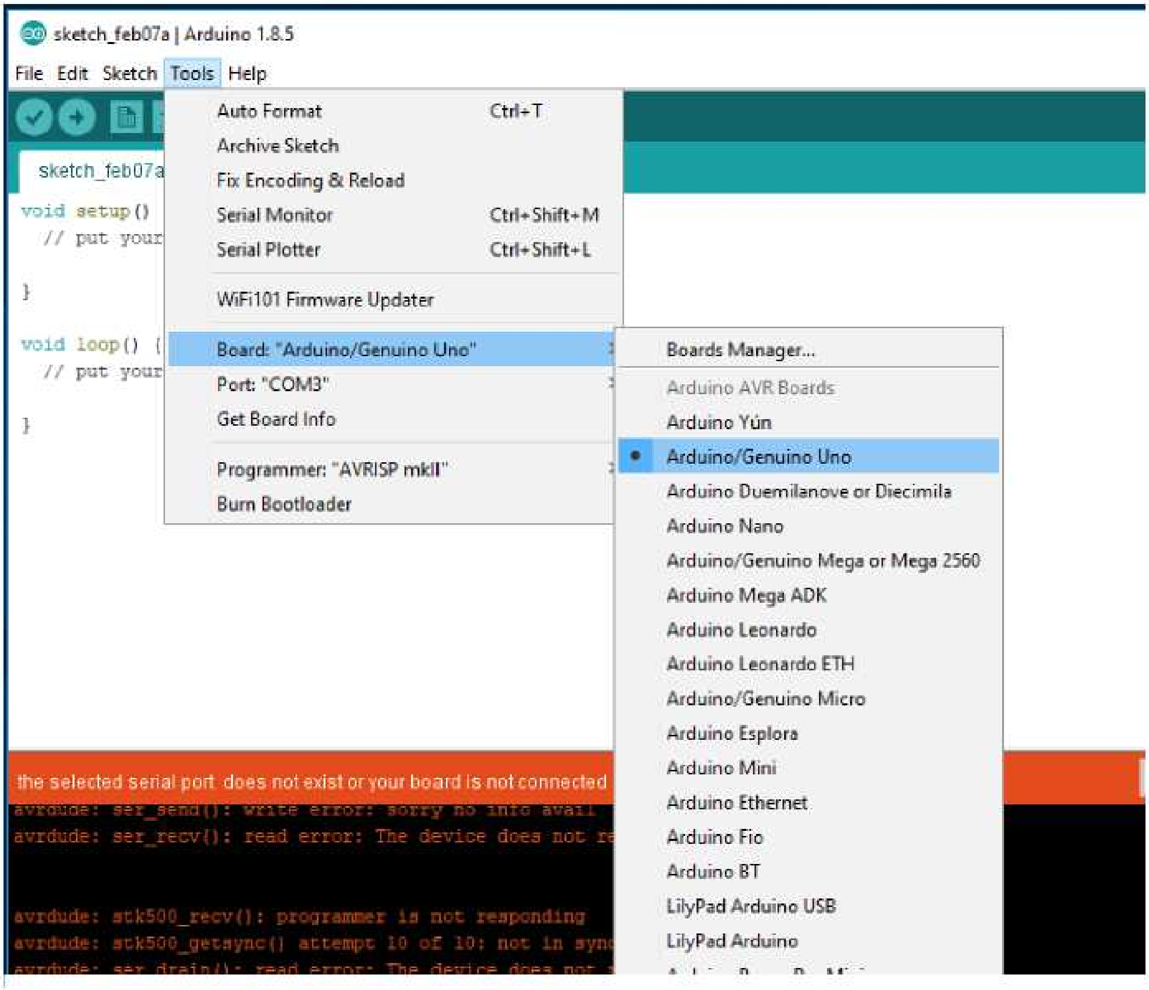
After installation, you should be greeted by default Arduino IDE window:


sketch_feb07a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_feb07a
Arduino/Genuino Uno on COM3

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig. 3 Arduino IDE window

To work with TotemDuino, first you must choose the correct board for which code will be compiled, as well serial port, by which the compiled firmware will be uploaded to it. You can select the board by clicking **Tools-> Board**. In the drop-down menu, please select **Arduino/Genuino UNO** board:



```
void setup() 
  // put your

}

void loop() {
  // put your

}
```

the selected serial port does not exist or your board is not connected

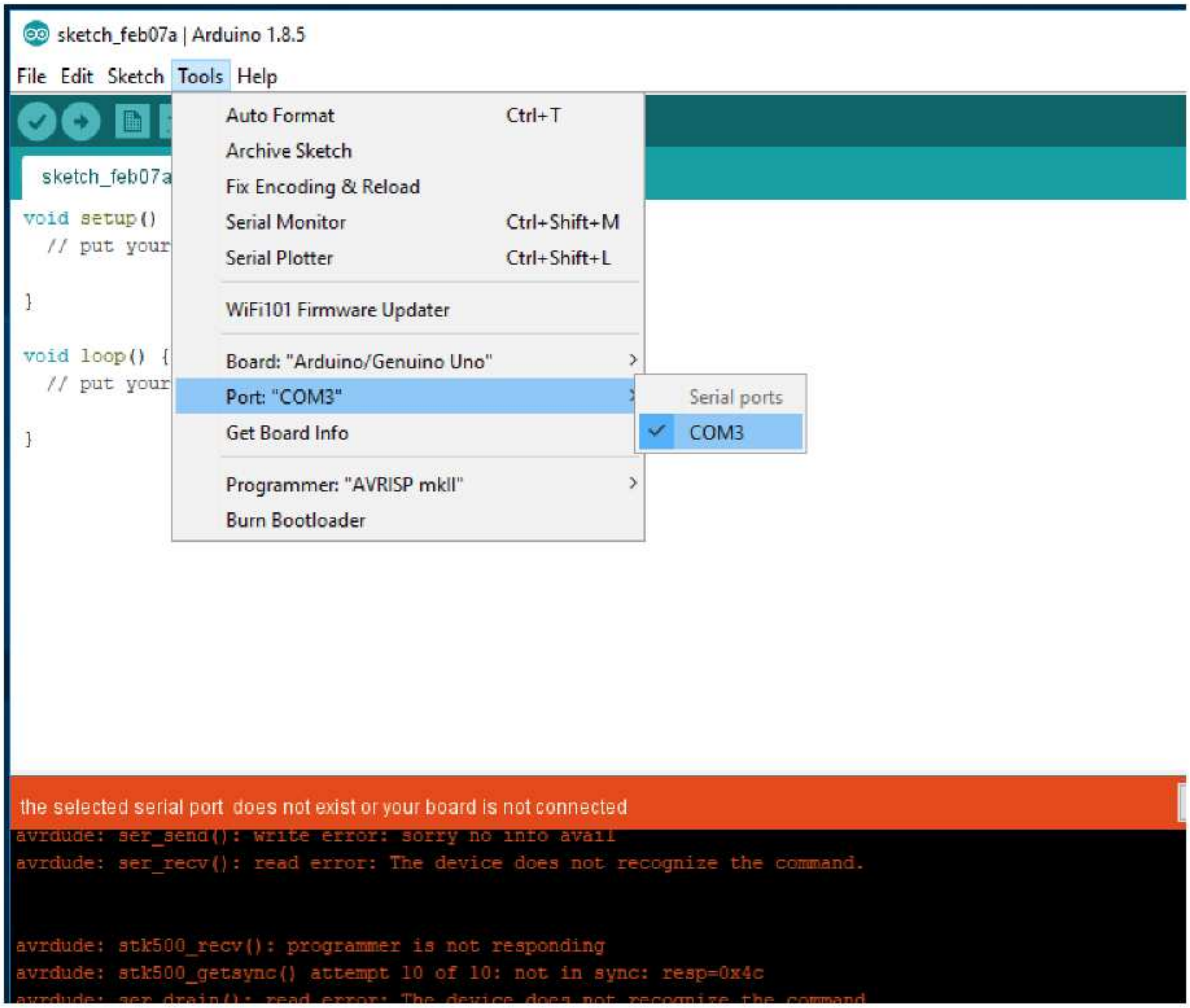
```
avrdude: ser_send(): write error: sorry no info avail
avrdude: ser_recv(): read error: The device does not re


avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 10 of 10: not in sync
avrdude: ser_drain(): read error: The device does not
```

Fig. 4 Board selection

Next, we'll select serial port. Because the number of available serial ports differ from one computer to another, and there's no way to know which port is the correct TotemDuino one, you might need to follow this procedure.

First check what ports are available when TotemDuino is disconnected, and compare the list once you connect TotemDuino back.

The newly appeared port will be the port that should be used. On most laptops there are no extra serial ports, so you might see TotemDuino port as being the only one available.


sketch_feb07a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_feb07a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM3"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
✓ COM3

```
void setup() {
  // put your
}

void loop() {
  // put your
}
```

the selected serial port does not exist or your board is not connected

```
avrdude: ser_send(): write error: sorry no info avail
avrdude: ser_recv(): read error: The device does not recognize the command.


avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x4c
avrdude: ser_drain(): read error: The device does not recognize the command
```

Fig. 5 Selecting TotemDuino serial port

Once that's done, we can start writing firmware for it. As always, its best to start with the basics, so first we should try to upload a simplest firmware which only blinks the onboard LED on the TotemDuino.

Using **File->Examples->01. Basics** drop-down menu select **Blink** example firmware:

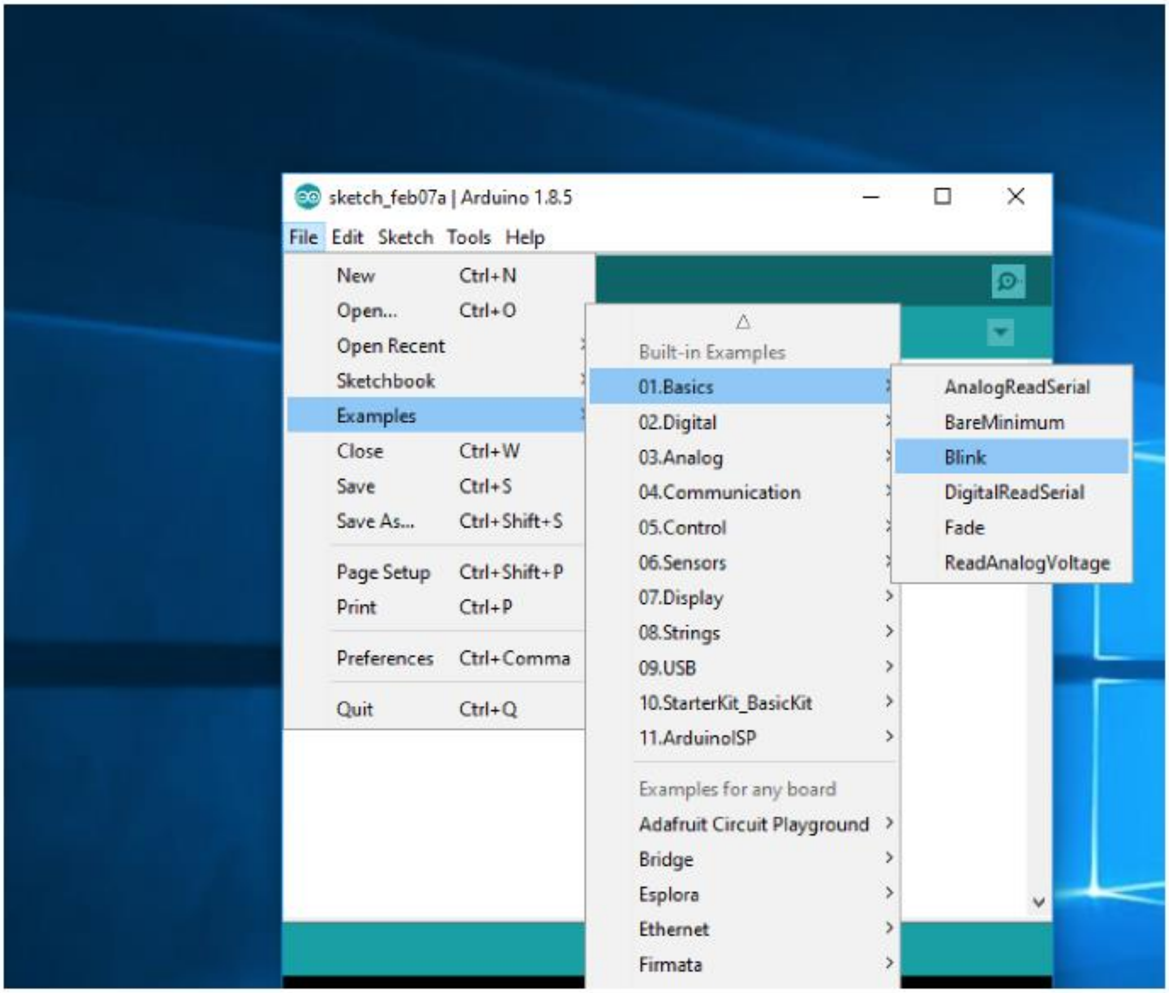
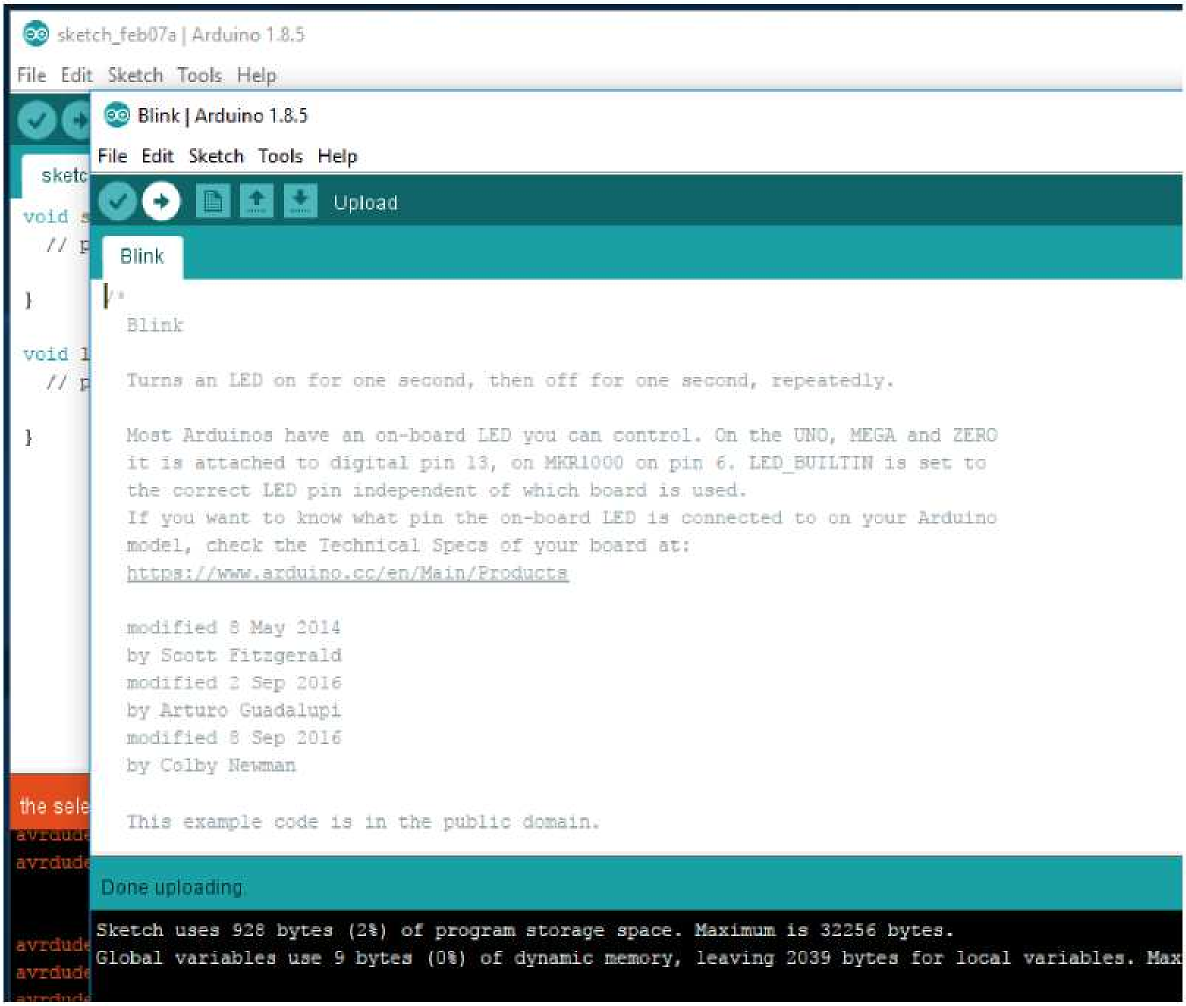


Fig. 6 Selecting example firmware

This will open a new Arduino IDE window with the source code of that firmware. Try to compile and write the firmware to TotemDuino by clicking **Upload** button (icon with right arrow):


sketch_feb07a | Arduino 1.8.5
File Edit Sketch Tools Help
Blink | Arduino 1.8.5
File Edit Sketch Tools Help
Upload
Blink

```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.
```

Done uploading.

```
Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Max
```

Fig. 7 Uploading firmware successfully

You should see a notification “Done uploading” in the bottom status bar. Now check if the firmware does what it’s supposed to:

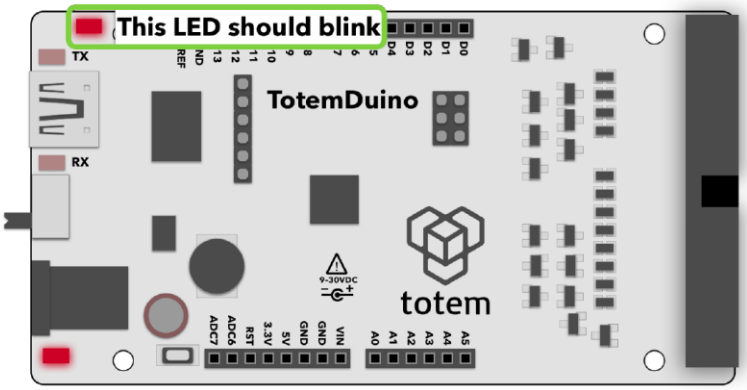


Fig. 8 Example LED blink firmware

If the LED blinks, you got every bit working! Now you can start diving into the world of programming by exploring other examples or creating your own firmwares for it. You could plug external sensors, shields to bring even the craziest ideas to life.

Have fun!

Technical specifications

Microcontroller: Atmega328P

Operating voltage: 5 V or 3.3 V (user selectable)

Input voltage: 9 - 30 V DC, 5 V USB

Maximum input current: 1A (DC), 500mA (USB)

Flash memory: 32 kB

SRAM: 2 kB

EEPROM memory: 1 kB

General purpose I/O pins: 14

Analog input pins: 8

Width: 150 mm

Height: 29 mm

Weight: 30 g