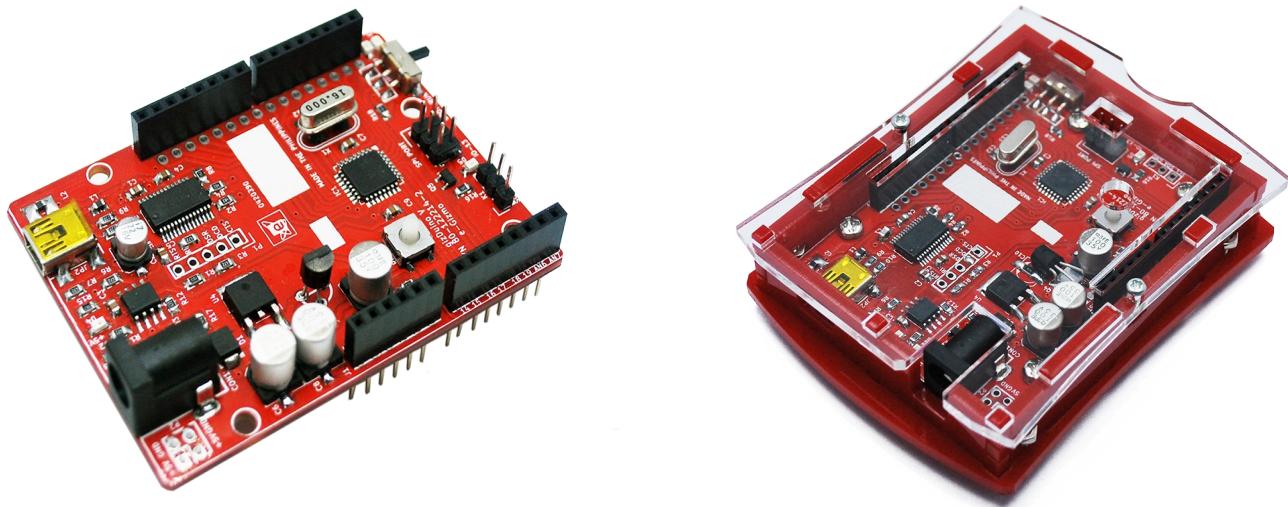


gizDuino 5 with ATmega328P



Technical Manual Rev 1r0

gizDuino 5 is the latest iteration of our continually evolving Arduino compatible series of programmable controllers. It provides added features suggested by users including (see page 3)



FEATURES:

- Arduino Compatible (Arduino UNO)
- Arduino IDE software
- 20 I/Os
- Prolific Driver (PL2303)
- USB Power source indicator
- Buffered UART TX/RX indicator
- UART switch
- Buffered I/O 13 indicator
- I2C port connections
- Additional +5V power port
- ICSP header connector
- Reset Button

GENERAL SPECIFICATIONS:

- **Input Supply:**
 - External Supply: 7V to 12VDC
 - USB power: +5VDC
- **Current Limit:** 2A
- **Current per pin:** 40mA
- **Digital I/Os:** 20 (D0-D19)
- **Analog Input:** 6 (A0-A5)
- **PWM pins:** D11,D10,D9,D6,D5,D3
- **Driver IC:** PL2303 Prolific
- **USB connector:** miniTypeB
- **USB cable:** miniTypeB to TypeA
- **Power source on-board:**
 - +5V , +3.3V DC , GND
 - +VIN = External Supply
- **PCB Dimensions:** 71mmx54mm

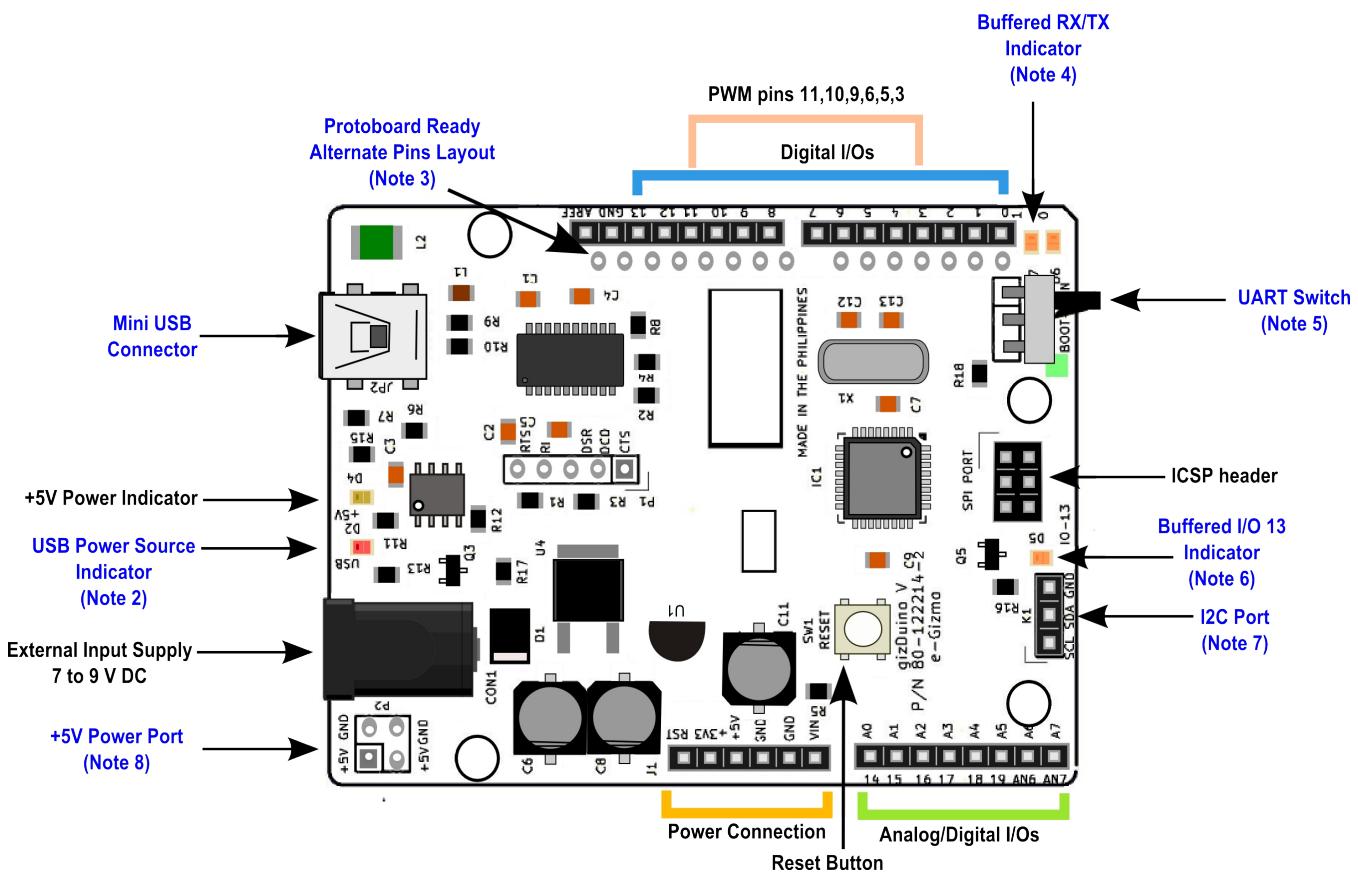


Figure 1. Major parts presentation of gizDuino Version 5.0 with ATmega328P

Important:

Install the latest **PL2303** driver in your Windows PC before connecting and using your gizDuino with your PC.

Components Specifications:

IC1 - ATMEGA 328P Microprocessor Chip

U1 - On-Board +3v3 VoltageRegulator

U2 - Supply Dual Operational Amplifier

U3 - PL-2303 USB to RS232 Bridge Controller

U4 - On-Board +5V Voltage Regulator

Table 1. Digital I/O Pin assignments

No.	I.D.	Description
1	AREF	Analog Reference
2	GND	Ground
3	13	Digital I/O (SCK)
4	12	Digital I/O (MISO)
5	11	Digital I/O (PWM/MOSI)
6	10	Digital I/O (PWM)
7	9	Digital I/O (PWM)
8	8	Digital I/O
9	7	Digital I/O
10	6	Digital I/O (PWM)
11	5	Digital I/O (PWM)
12	4	Digital I/O
13	3	Digital I/O (PWM)
14	2	Digital I/O
15	1	Digital I/O (TX)
16	0	Digital I/O (RX)

Tips and Reminders:

Pin 13 is also the pin used by the data transfer LED Indicator.

Table 2. Analog/Digital I/O Pin assignments

No.	I.D.	Description
1	A0	Analog I/O or Digital I/O
2	A1	Analog I/O or Digital I/O
3	A2	Analog I/O or Digital I/O
4	A3	Analog I/O or Digital I/O
5	A4	Analog I/O (SDA) or Digital I/O
6	A5	Analog I/O (SCL) or Digital I/O

Tips and Reminders:

For I2C interface, A4 (SDA) and A5 (SCL) can be used.

In 328P, A6 and A7 has no connections from the microcontroller.

Table 3. Power pin assignments

No.	I.D.	Description
1	Reset	Reset
2	+3.3V	3.3VDC Device Power supply
3	+5V	5VDC Device Power supply
4	GND	Ground
5	GND	Ground
6	Vin	+5V(USB) / Vin(Ext)

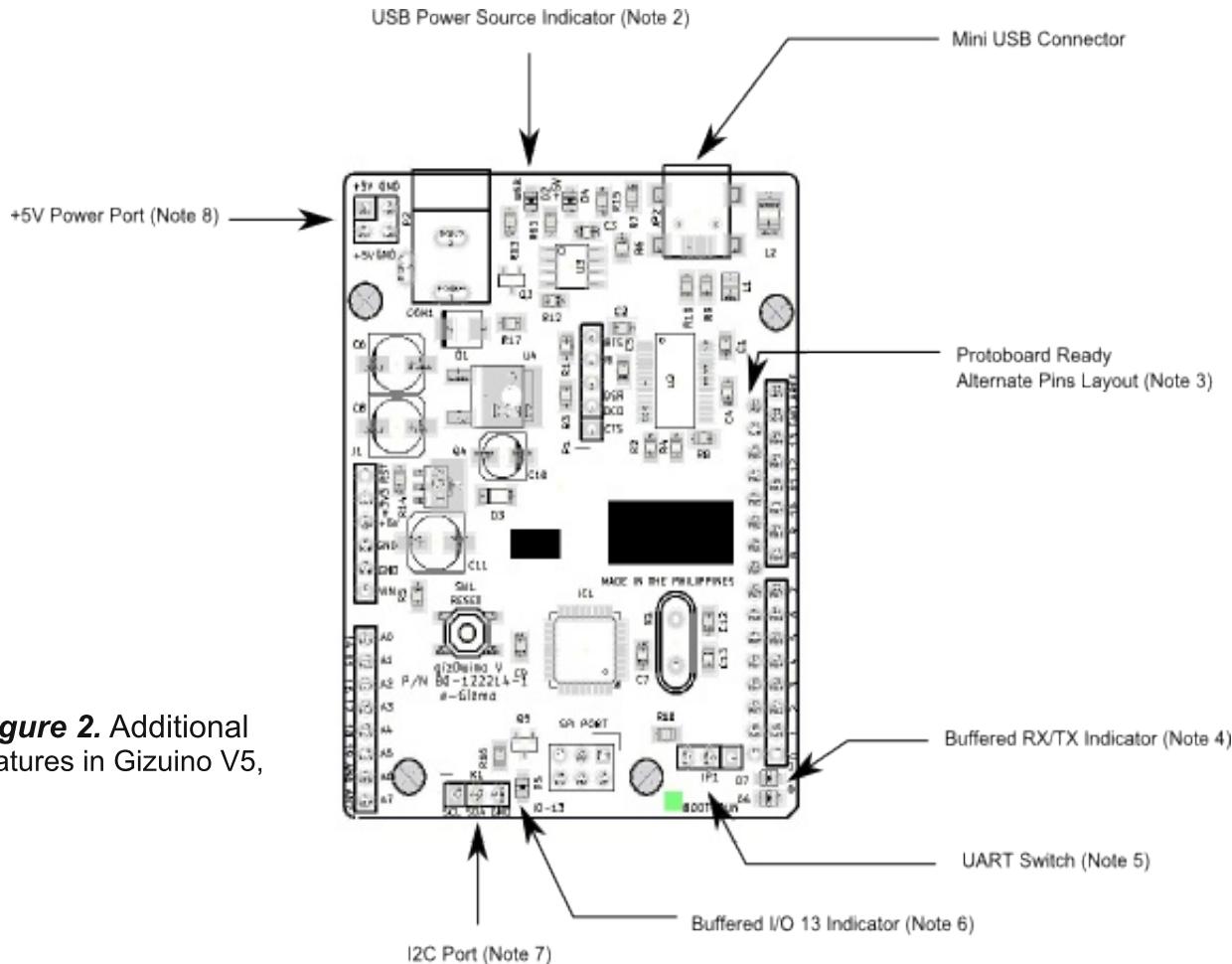


Figure 2. Additional features in Gizuino V5,

NOTE 1

Robust reset circuitry that does its jobs more reliably. Early variants of gizDuino used similar reset circuit as employed in the Arduino Diecemilla and inherited the problems that went with. The reset circuit can be easily overwhelmed by the addition of shields, causing frequent connections problem with the PC during the downloading process. The new reset circuit is expected to minimize that.

NOTE 2

USB Power Source indicator - Tells you when power is drawn from the USB port.

NOTE 3

Protoboard Ready alternate pin layout - At your option, you can solder additional header on this pins. This will allow you to mount and connect your gizDuino5 board to a standard perforated prototyping PCBs.

NOTE 4

Buffered UART TX/RX indicator - Now a visual means to monitor the activity of one of the most used port of your gizDuino. LEDs flashes when data (of sufficient duration) flows through these pins.

NOTE 5

UART Switch - A shield or extension device that connects through your gizDuino UART port can interfere with the program downloading process, resulting in those "STK-500" failures. The usual workaround is to remove the shields or expansions during program downloading, and reseat them later after program transfer is completed. With the UART switch, you don't have to go through this inconvenience. Just switch to "BOOT" position during program transfer, and switch back to "RUN" position after program downloading is completed.

NOTE 6

Buffered I/O 13 indicator - Frees your I/O 13 from the LED burden. Now you can even use the internal pull ups of the I/O13 and use it as if the LED is not there.

NOTE 7

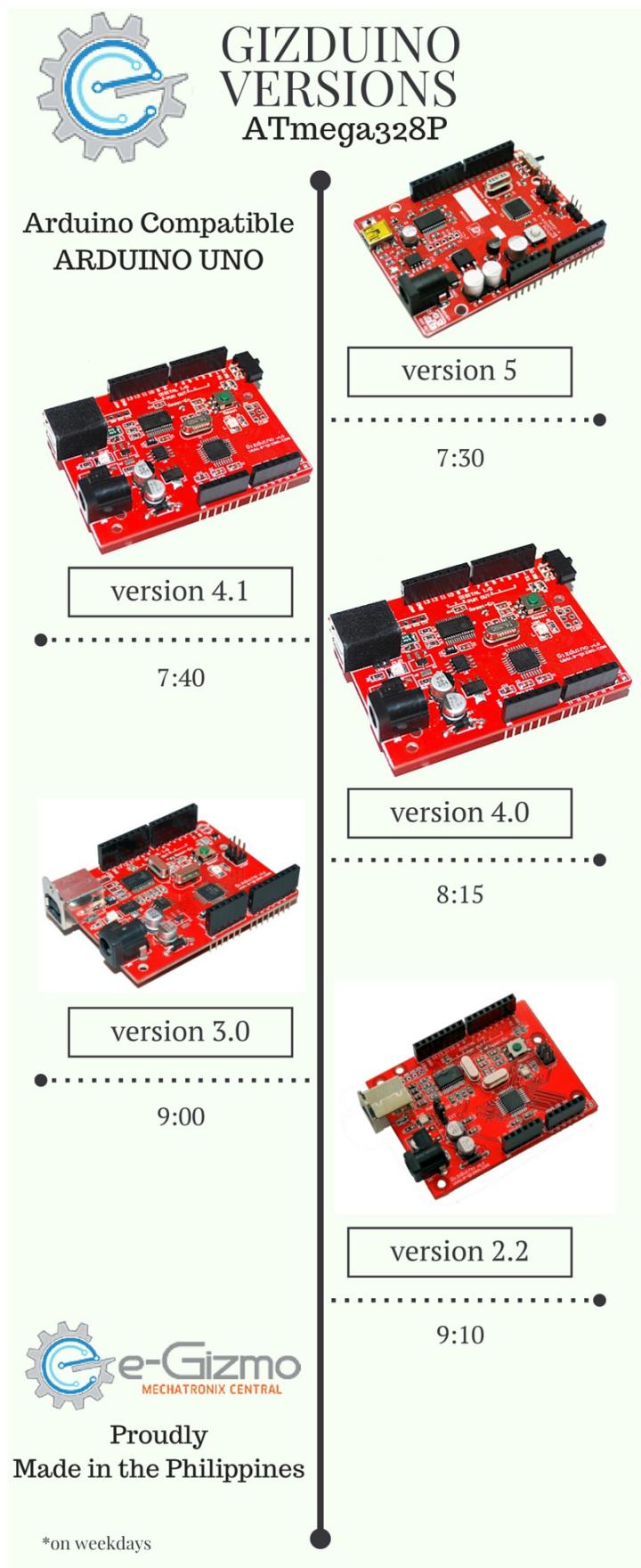
A Separate I2C port to make connecting to your I2C devices even simpler and easier.

NOTE 8

Additional +5V Power Port. More +5V to GND take off points.

GizDuino Version History

The timeline of gizDuino versions, differences shown in figure at the right. From version2.2 to the newest release of version 5.



Gizduino Version 5 is an ATmega328P IC microcontroller on board. To use this you need a software for programming. Arduino IDE is an open software that is compatible on gizduino boards.

SETTING UP

1. Install the Arduino IDE.

- You can download the Arduino IDE @wwwarduino.cc/Main/Software. (Add the gizDuino patch file) to get the Gizduino board list. (See the figure below) OR Go to eGizmo website, Click "Downloads". Browse 'G'. Click the arduino version for your OS.

For Arduino 1.0.6 version below:

Go to eGizmo Website @www.e-gizmo.com. Click "Downloads". click the Gizduino patch 1.0.6 and below.

For Arduino 1.6.7 and up. Dowload the Gizduino patch 1.5.x and 1.6.x up.

To add the gizDuino patch files:

Copy the "gizduino-" folder. Place it to Arduino folder>hardware folder>gizduino- (paste it).

Done!

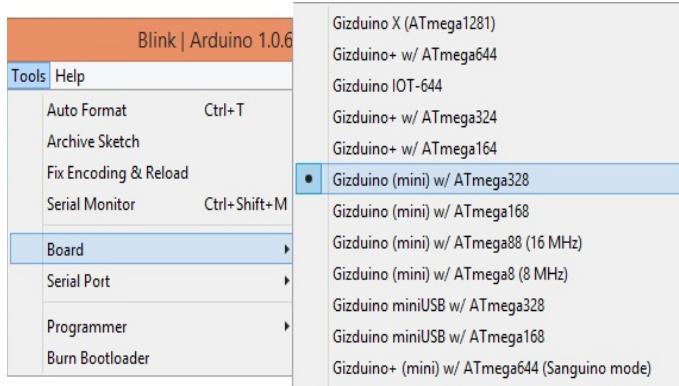


Figure 3. Gizdino Board list.

2. Install the Prolific Driver.

- You can download the Prolific Driver (PL2303) for gizduino driver.

@www.prolific.com.tw/US>ShowProduct.aspx?pid=225&pcid=41 OR if you already download the Arduino IDE in eGizmo website. Just go to Arduino folder>Drivers folder>Prolific driver folder>Install the PL2303.exe.

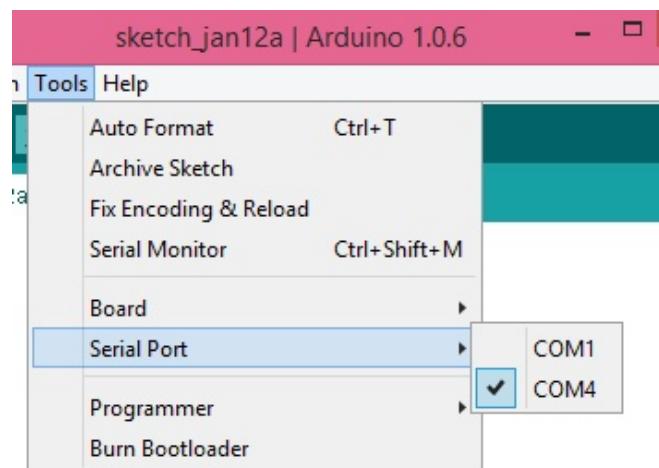


Figure 4. Com port Select.

Make sure the installation is completed. Then Connect the USB cable from gizduino V5 to PC and to appear the COM number.

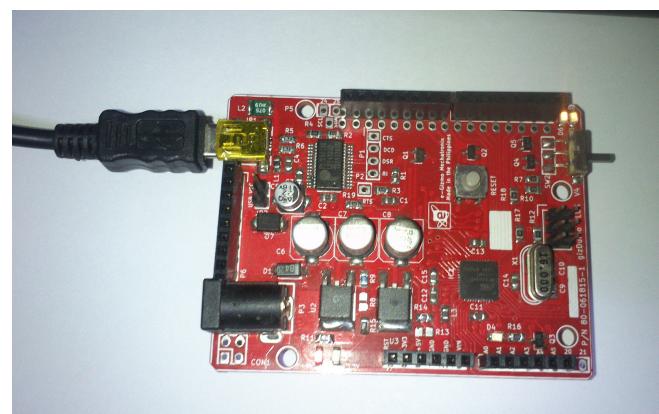


Figure 5. USB connections,



OPEN ARDUINO.EXE

Select Board:

Go to TOOLS>BOARD> SELECT "Gizduino (mini) with ATmega328P" or "Arduino Uno"

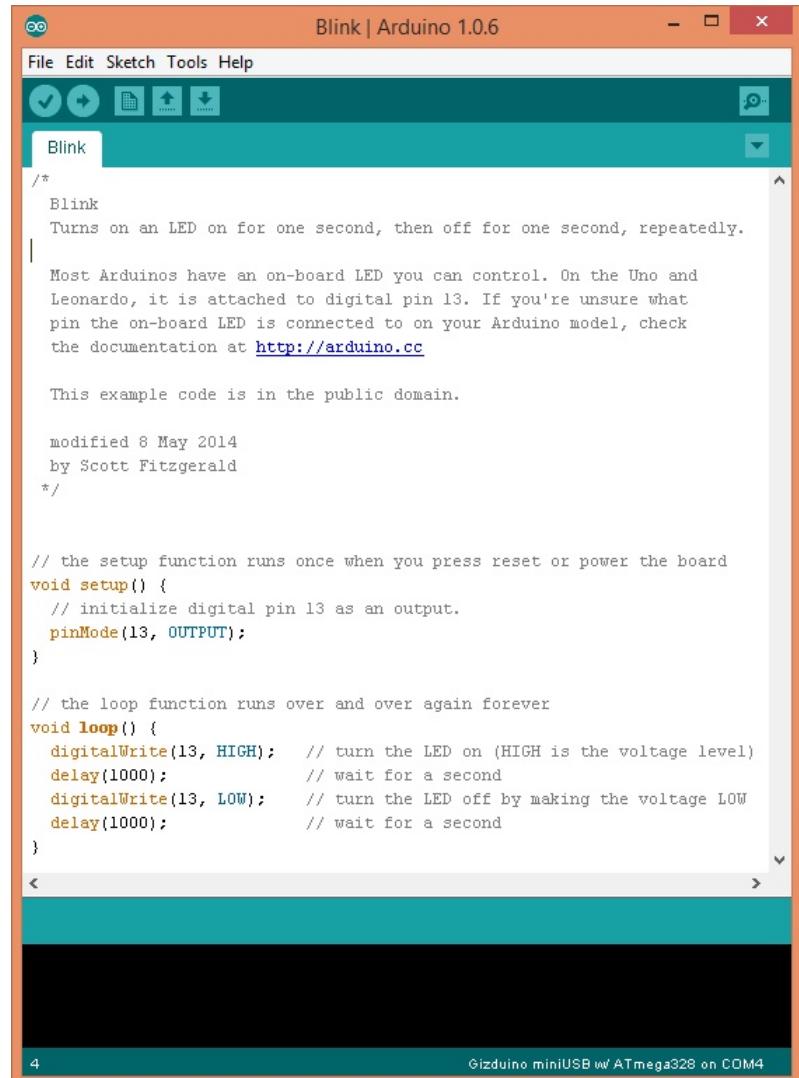
Select Comport:

Go to TOOLS>SERIALPORT> SELECT "COM#"

For Blink Test:

Go to FILE>EXAMPLES>1.BASICS>BLINK

Then Click Upload.



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0.6". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for Save, Undo, Redo, Open, and others. The sketch name "Blink" is selected in the dropdown menu. The code editor contains the following code:

```
/*
Blink
Turns on 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 and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://arduino.cc

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

The status bar at the bottom shows "4" and "Gizduino miniUSB w/ ATmega328 on COM4".

Figure 6. Arduino IDE.

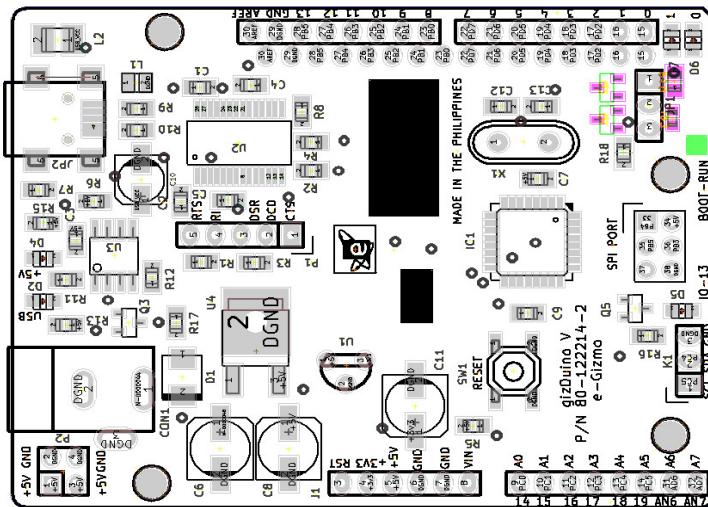


Figure 7. Silkscreen layout Guide

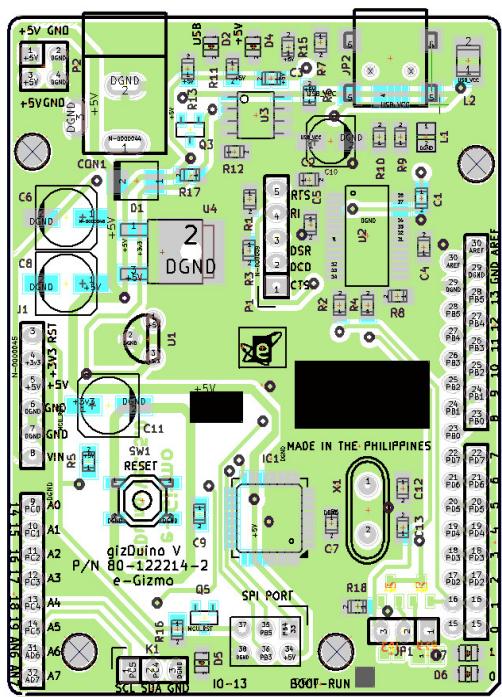


Figure 8. Bottom PCB layout guide

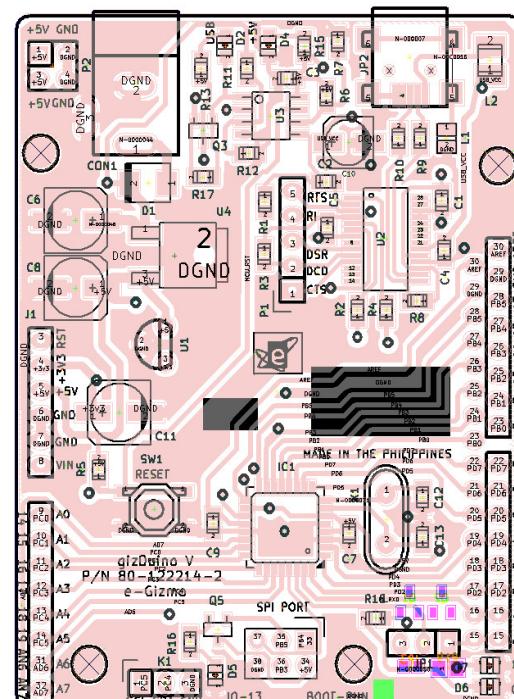


Figure 9. Top PCB layout guide

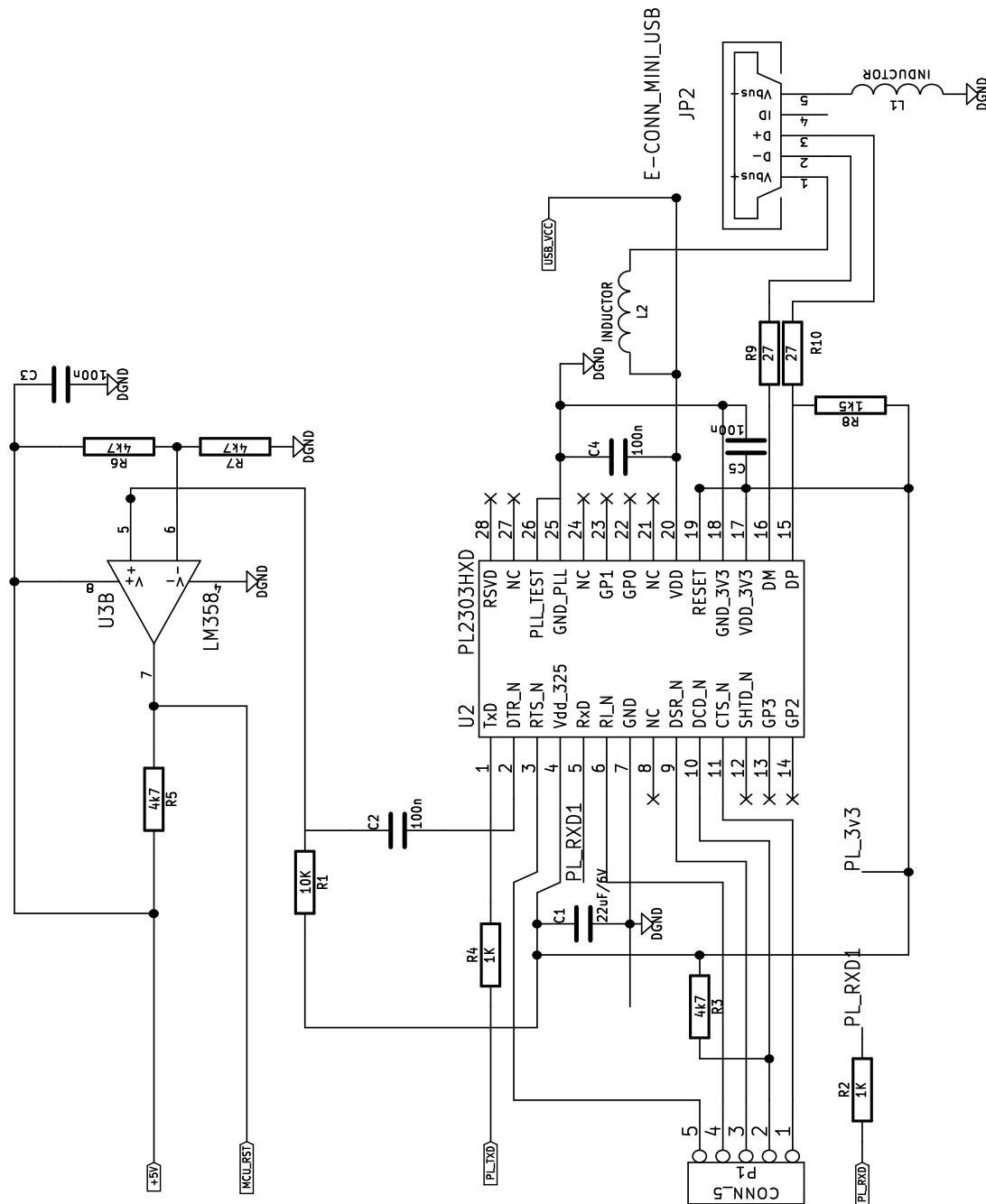


Figure 10. Schematic Diagram of e-Gizmo Prolific Driver PL2303.

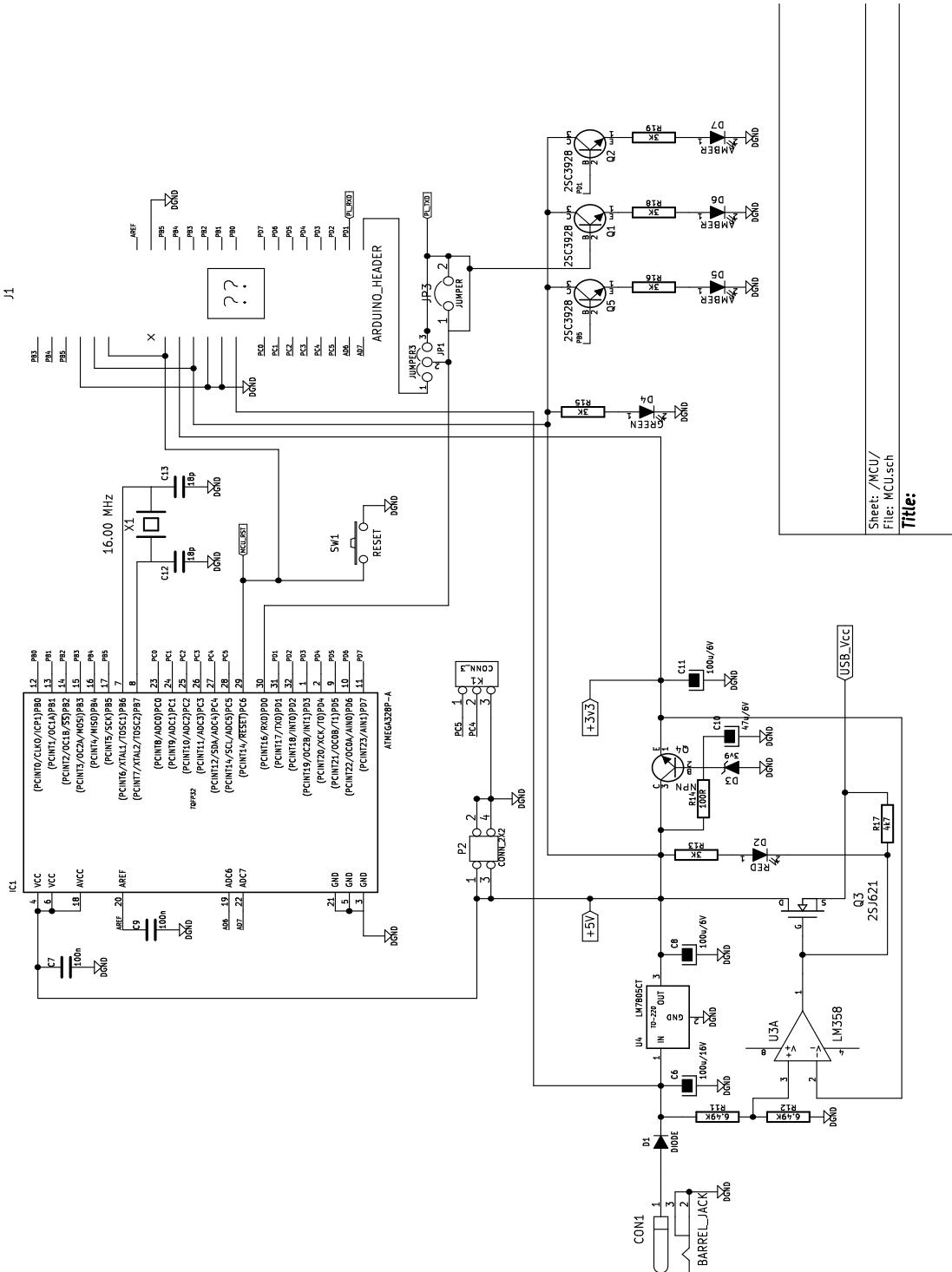


Figure 11. Schematic Diagram of e-Gizmo Gizduino V5 (ATmega328P).