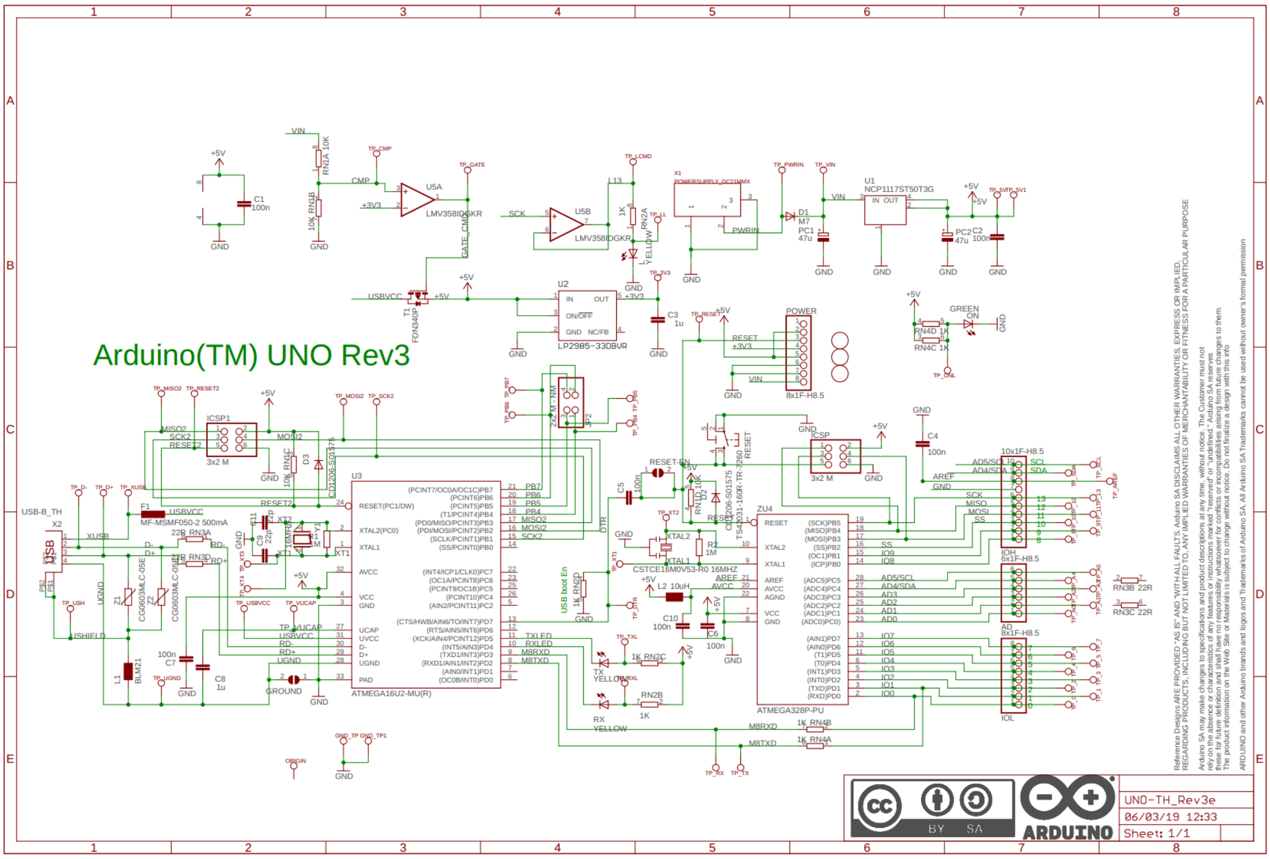
Arduino Uno

For the realization of the infusion pump control system, we use Arduino Uno. For the functions that our device is to perform, all the electronic components of Arduino one are not needed. This section provides an explanation of the selection or elimination of these electronic components.

Arduino Uno Schematic



Arduino Uno consists of a **main integrated** (**ATMEGA328P-PU**) and a **secondary integrated** (**ATMEGA16U2-MU(R)**). The ATMEGA16U2-MU(R) integrated that handles USB communication with a serial interface consisting of the TXLED RXLED signals, communicates with the main integrated for sketch loading. Both integrates operate at 5V, and the working frequency of the 328P IC is generated through a 16 MHz ceramic resonator.

- The **INPUT/OUTPUT PINS** of the main integrated are directly connected with a set of 2.54 mm pin-headers; there are no external protections. Although each microcontroller has an internal protection realized by two built-in diodes for each INPUT/OUTPUT PIN in the integrated circuit
  The datasheet recommends that each INPUT/OUTPUT PIN should not have a voltage greater than 0.5V applied to it than the supply voltage of the integrated.

- The **RESET PIN** is connected to a pull-up resistor to a diode, both connected to 5V, the PIN in turn is connected to a pushbutton and the ICSP programming connector, then the PIN through another capacitor is tied to a pin of the secondary integrated, this is to allow the auto-reset function. This function allows the sketch to start immediately after loading without having to press the reset button or without having to power the board again.

- Through a series of connections, you have a connection between the secondary integrated and the USB port. Among these connections we recognize the following elements:

    - The **piezoelectric** we use to manage the clock frequency of the secondary integrated. The piezoelectric you have to ground it through two capacitors.

    - The **two variable resistors Z1** and **Z2** allow you to regulate the voltage provided by the USB input to the second integrated (you must always provide a voltage of 5V so as not to burn out the integrated).

Arduino Uno can be powered via a **USB port** or via an **external connector**.

- In the external connector there is a linear regulator that when powered provides a voltage of 5V. *The NCP linear regulator can deliver a maximum current of 1A, but the maximum current that the integrated can deliver is referred to the minimum voltage, i.e., 6V, because dissipation aspects must be taken into account. Basically, the integrated to not go into thermal protection can deliver a maximum current of 1A when the supply voltage is 6V, and as the supply voltage goes up the maximum current delivered by the integrated without going into thermal protection will be less and less.*

- Arduino Uno has a **scheme for choosing the supply voltage**.

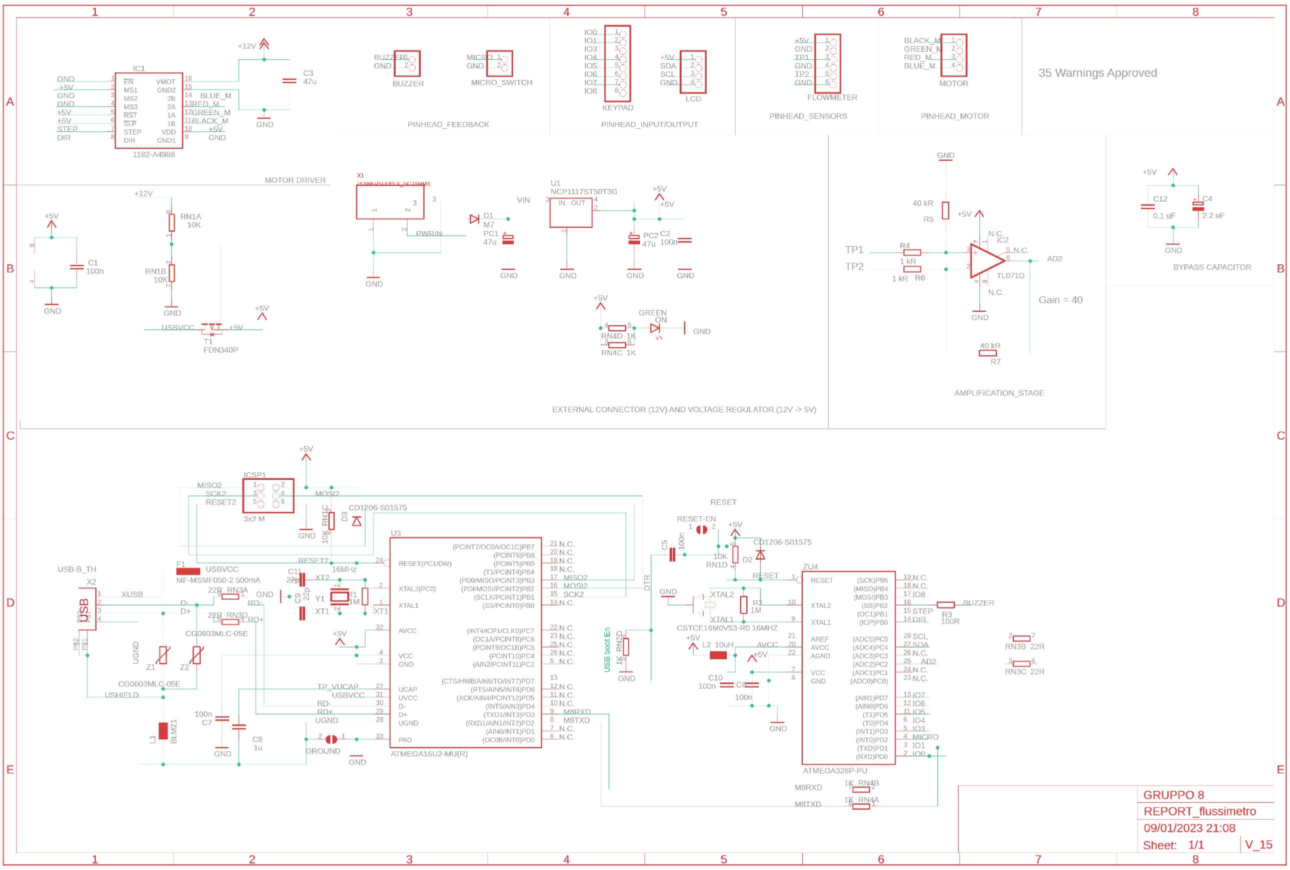
  When the external VIN voltage is present the comparator output will be at logic high level, the P-type mosfet will be off, and the +5V voltage will be determined only by the linear regulator.

  While when only the USB voltage is present the comparator output will be at logic low level, so the mosfet will be on and the USB supply voltage will also be the +5V voltage and consequently the board supply voltage (USB Vcc).

  We have another linear regulator that generates the 3.3V voltage for external use.

There is an **LED** on the board that indicates that the Arduino is turned on.

Arduino Uno Schematic for the project



For the realization of the infusion pump we need both the **USB port** and the **external controller**.

In the **scheme for choosing the supply voltage** to provide to the Arduino, we can eliminate the linear regulator that provides the 3.3 V output, because all the sensors and electronic devices we are going to use will be supplied with 5V.

We can eliminate the other **LEDs** on the Arduino board, leaving only the LED that indicates its power on.

Regarding the **Reset**, we can remove the external button, leaving only the auto-reset function explained in the previous section. Consequently, you can eliminate ICSP programming connector, but not ICSP1 programming connector

Pins PD3 and PD2 of the secondary integrated are the **TXD** and **RXD** pins that we leave for receive-transmit communication (full-duplex).

All Arduino **pinheads** have been eliminated and new pinheads have been inserted for each electronic component that you need to connect with Arduino:

- Pinhead feedback:
  - Pinhead for the buzzer
  - Pinhead for the micro-switch
- Pinhead input/output
  - Pinhead for the keypad

    - Pinhead for the LCD screen
- Pinhead sensors
    - Pinhead for the flow sensor
- Pinhead motor
    - Pinhead for the motor NEMA17

The **motor driver** was inserted directly on the Arduino board.

An **amplification state** is provided for the analogical output of the FS1012 flowmeter. The output of this sensor is in mV, you want to get an output in V. You use a differential amplifier with gain equal to 40. A gain of 40 was chosen because you need to maintain an amplifier output below 5 V. In fact, you need to stay below 4.5 V because already at this voltage the amplifier fed between 0 and 5V starts to saturate. To obtain a gain of 40, the following relationship is used:

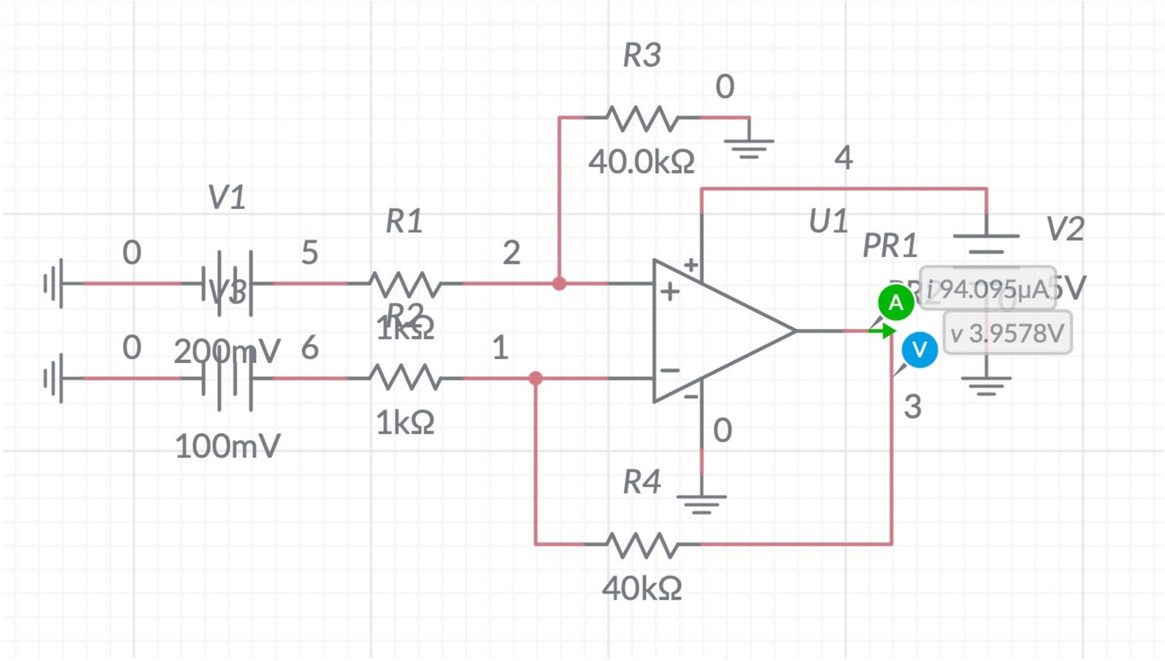
$$Gain = \frac{V_{out}}{V_{in}} = \frac{V_{out}}{(V_2 - V_1)} = \frac{R_2}{R_1} = 40$$

the values chosen for the two resistors are:

- $R_2 = 40\ \mathrm{k\Omega}$
- $R_1 = 1\ \mathrm{k\Omega}$

The amplifier used is a TL071D.

Bypass capacitors for the amplifier's 5V supply were built into the amplifier stage.


R3
0
40.0kΩ
4
V1
R1
U1
PR1
V2
0
5
2
V3
5V
A
i 94.095μA
v 3.9578V
V
R2
1kΩ
0
200mV
6
1
3
1kΩ
100mV
0
R4
40kΩ