

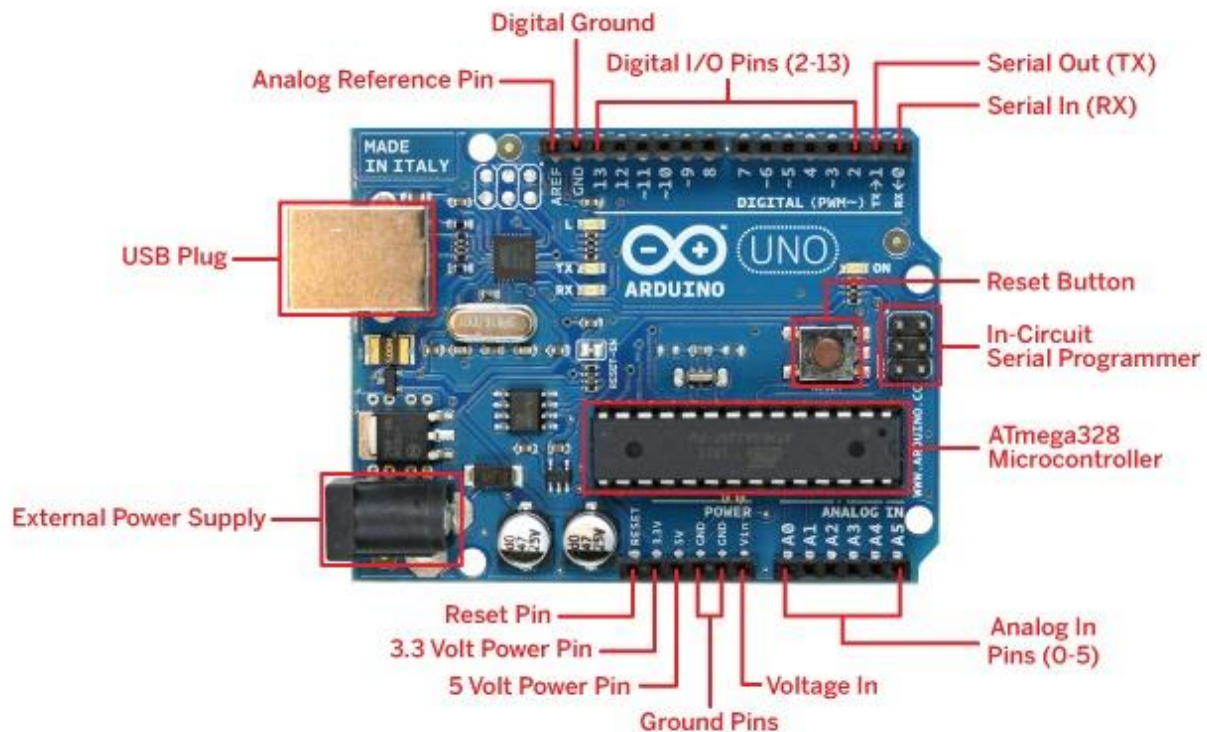
Board Uno Specifications:

Technical specs

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

SOURCE: <https://www.arduino.cc/en/Main/ArduinoBoardUno>

Our team member H. Wu has plotted the following schematics:



Pins used:

Pins 6, 7 used for the fingerprint scanner.

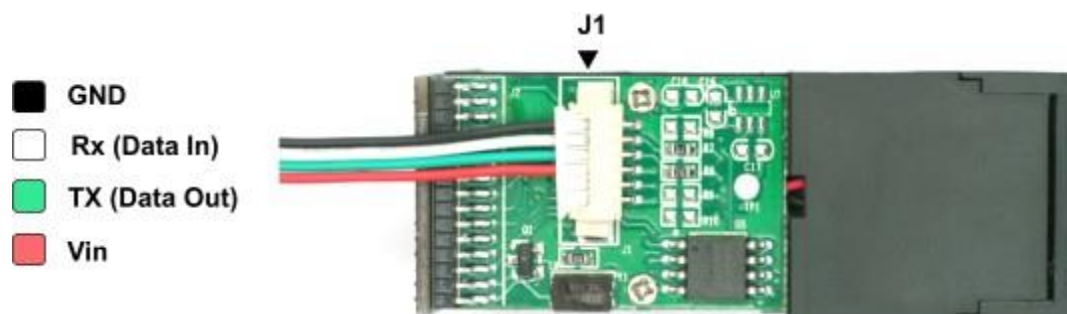
Prints 12, 11, 5, 4, 3, 2 used for crystal LCD screen.

Pings 8, 9, 10, 13 used for input press buttons.

Pins not being used:

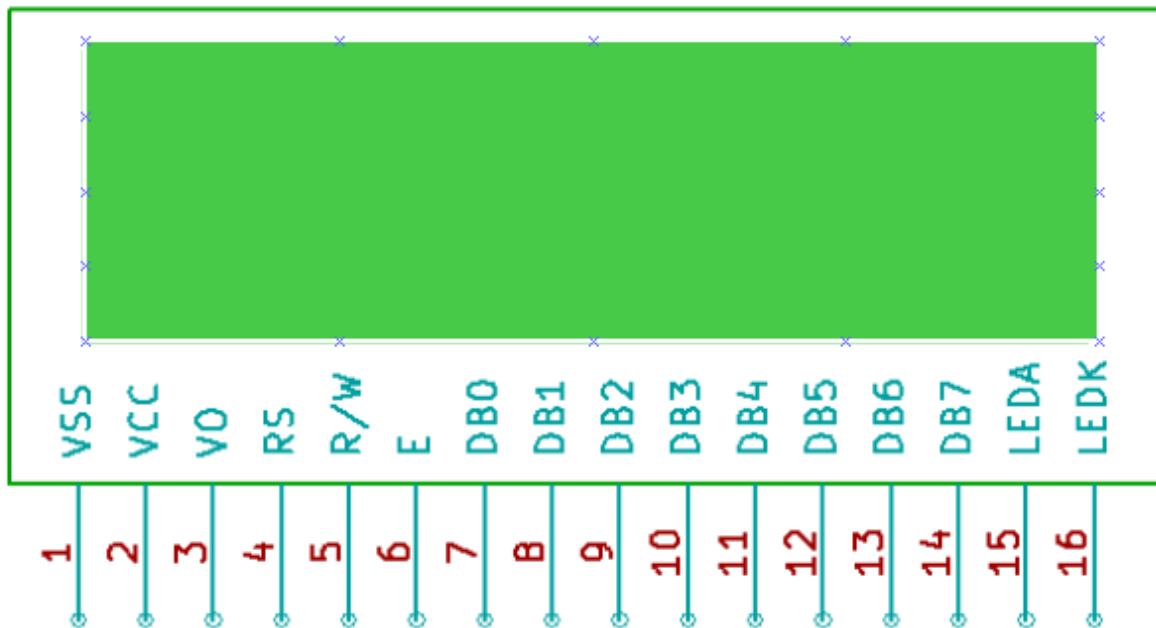
Pin 0, 1 and other analog pins because we are not using any potentiometers, or pots.

Lastly, ping 5v and Gnd (ground) are used to provide power for all devices.



(This picture is from the bottom of the scanner)

LCD_2X16_SIL



Connecting together:

Arduino Pins	Destination Pins
2	16x2 Crystal LCD pin 14 (DB7)
3	16x2 Crystal LCD pin 13 (DB6)
4	16x2 Crystal LCD pin 12 (DB5)
5	16x2 Crystal LCD pin 11 (DB4)
6	Finger Sensor (TX) Reads data from sensor
7	Button 3 (Decrement Button)
8	Button 2 (Confirm Selection)
9	Button 4 (Increment Button)
10	Finger Sensor (Rx) Sends data to sensor
11	16x2 Crystal LCD pin 4 (RS)
12	16x2 Crystal LCD pin 6 (E)
13	Button 1 (Change mode)

Sensor Pins	Destination Pins
GND	Connects to Arduino Chip Ground Pin
Rx	Connects to Arduino Pin 6 to receive data
TX	Connects to Arduino Pin 7 to send data
Vin	Connects Arduino 5V Power Pin

LCD Pins	Destination Pins
1	Connects to Ground
2	Connects to 5V Power
3	Connects to Ground with 220 Ohm resistor
4	Connects to Arduino Pin 12
5	Connects to Ground
6	Connects to Arduino Pin 11
7	NULL
8	NULL
9	NULL
10	NULL
11	Connects to Arduino Pin 5
12	Connects to Arduino Pin 4
13	Connects to Arduino Pin 3 (PWM)
14	Connects to Arduino Pin 2
15	Connects to 5V Power Pin with 220 Ohm resistor
16	Connects to Ground Pin

Our team member H. Wu has programmed the Arduino Uno microcontroller to behave as the following:

Scan Mode - scans for any finger that comes in contact with the scanner.

Insert Mode - adds a new finger to the scanner's memory to be recognized.

Delete Mode - deletes a finger that is already being recognized by the scanner.

He also added the arduino with four customized buttons and programmed these buttons to do the following:

Button 1 - Switches between the three modes

Button 2 - Confirms selection/mode

Button 3 - Decrement finger ID (for insert mode and delete mode only)

Button 4 - Increment finger ID (for insert mode and delete mode only)

Due to the lack of pins provided by Arduino Uno, only 4 buttons can be implemented in this device. Other pins are dedicated for the scanner and the attached crystal-lcd screen. Only 5 finger ID were implemented in the code for fast rotation with only two buttons (button 3 and 4).

NOTE: Several libraries were including such as Bounce.h, LiquidCrystal.h, Adafruit_Fingerprint.h and SoftwareSerial.h

The LiquidCrystal.h library allowed me to print to the crystal lcd with easy, simply call their functions to print, wipe and change values freely.

Adafruit_Fingerprint.h is the most important library in this project, it allows me to program the scanner to do tasks as desired.

NOTE: The bounce.h was modified because it didn't work from the original package.
//lines added in the bounce.cpp

```
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include "Bounce.h"
```

This library is crucial for inputs such as button pressing and switches. This function basically sets a delay in milliseconds when it detects user's input. By setting the delay, the microcontroller ignores the initial fluctuation when the button is pressed, so the controller reads a smooth input.

NOTE:

Please read enroll.ino for detailed explanation of the code.

For further information about the device build or code, you may contact
haifeng.wu@jjay.cuny.edu