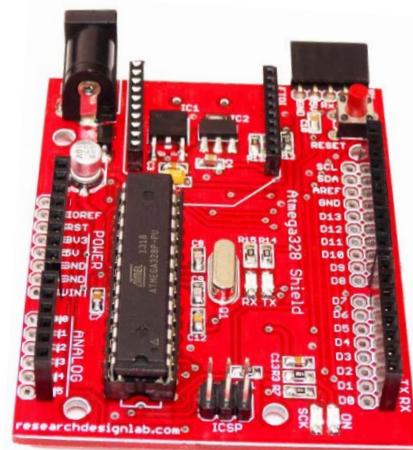




RDL-UNO ATmega 328



Development Board



Table of Contents

Overview	4
Features	4
Circuit Diagram	6
Difference between UNO R3 and RDL UNO ATMEGA 328	7
ATmega 328P	8
Features	8
Pin Configuration	9
Circuit Diagram	9
Application interface.....	10
LCD-SCREW SHIELD	10
XBee interface.....	13
ARDUINO EXPERIMENT BOOK	16
ARDUINO IDE	16

Overview

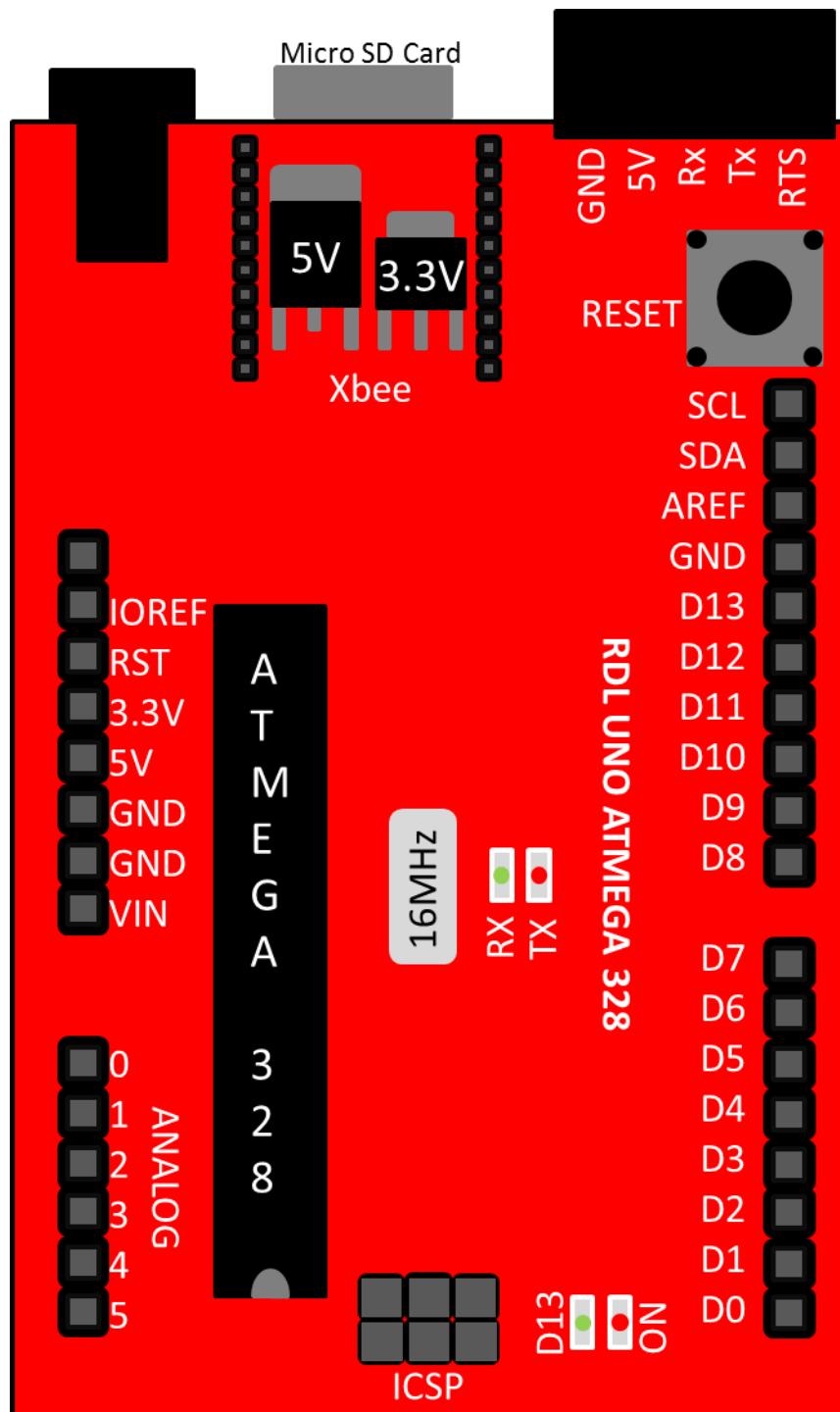
ATmega 328 is one of the most commonly used Micro controllers with open source platform amongst many hobbyist and industrial communities. The simplicity and the low power of ATmega 328 helps design many prototype boards which could be used in numerous applications.

We introduced RDL UNO ATmega 328 development board having common UNO interface to the available shields. The board also has inbuilt Xbee footprint with memory card interface in a single compact board without the need of doing external connections and hence as a whole the final product becomes cost effective. The RDL UNO ATmega 328 also includes 6 analog inputs, 14 digital I/O pins (6 amongst these could be used as PWM outputs), a crystal oscillator with 16 MHz frequency, a 5 pin female connector to connect FT232 plugin, a power jack, a reset button and an ICSP header. This board does not have an inbuilt USB to directly load the program from the computer. As we all know, once the program has been dumped into the board, the part of the USB connectors become useless. So to load the program we have given common interface through FT232 breakthrough board where we can connect to a computer and load the codes into the RDL UNO ATmega 328. Since the USB interface has been removed, we have provided Xbee foot print and Micro SD card slot which is commonly used by many electronics enthusiasts. Once the program has been dumped into the RDL UNO ATmega 328 we can remove the FT232 plugin and could be used independently for other purposes. This board is fully compatible with Arduino like open source software.

Features

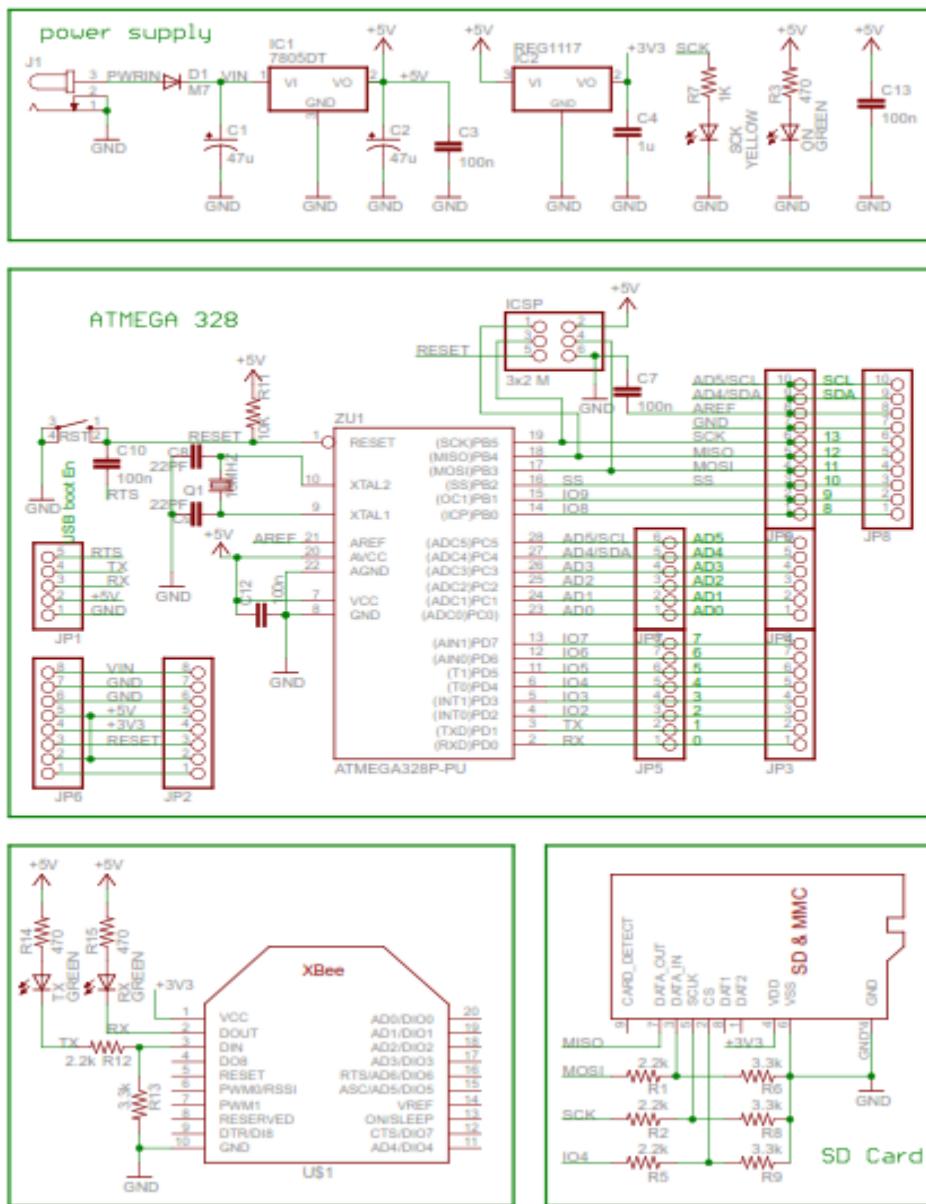
- ATmega 328P- controller
- Standard I/O interface for UNO shield.
- Xbee foot print for RF/WiFi communication.
- Inbuilt Micro SD memory card slot for Data logging application.
- Programmable with Arduino open source.

Application diagram

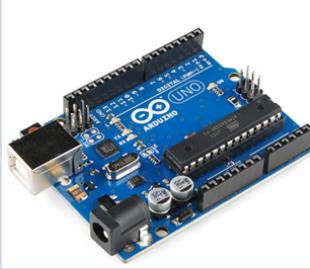
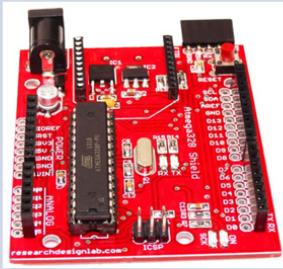




Circuit Diagram



**Difference between UNO R3 and RDL UNO ATMEGA 328**

Features	UNO R3/Freed UNO	RDL UNO ATMEGA 328
Image		
Standard I/O	✓	✓
Xbee Footprint for RF & Wi-Fi Communication	✗	✓
Micro SD slot	✗	✓
FT232 high speed USB to UART bridge	✗	✓
Inbuilt 5V & 3.3 V	✓	✓
Inbuilt Bootloader	✓	✓
Inbuilt USB to Serial	✓	✗
Swappable USB	✗	✓



ATmega 328P

ATmega328P is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino Uno

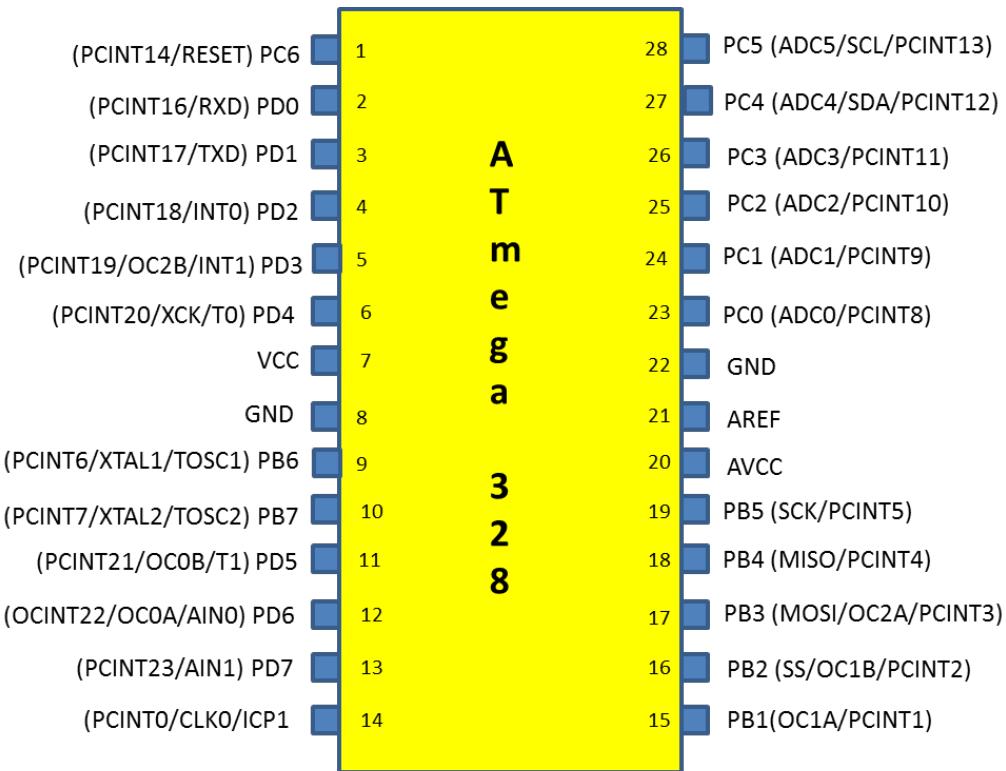


Features

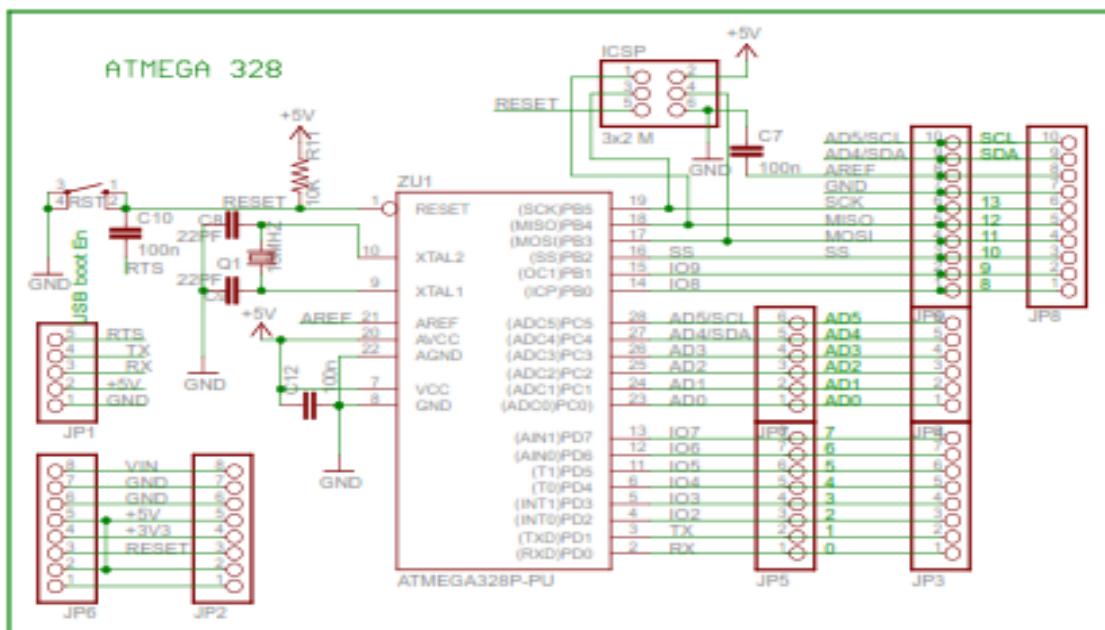
- 28-pin AVR Microcontroller
- Flash Program Memory: 32 Kbytes
- EEPROM Data Memory: 1 kbytes
- SRAM Data Memory: 2 kbytes
- I/O Pins: 23
- Timers: Two 8-bit / One 16-bit
- A/D Converter: 10-bit Six Channel
- PWM: Six Channels
- RTC: Yes with Separate Oscillator
- MSSP: SPI and I²C Master and Slave Support
- USART: Yes
- External Oscillator: up to 20MHz



Pin Configuration



Circuit Diagram



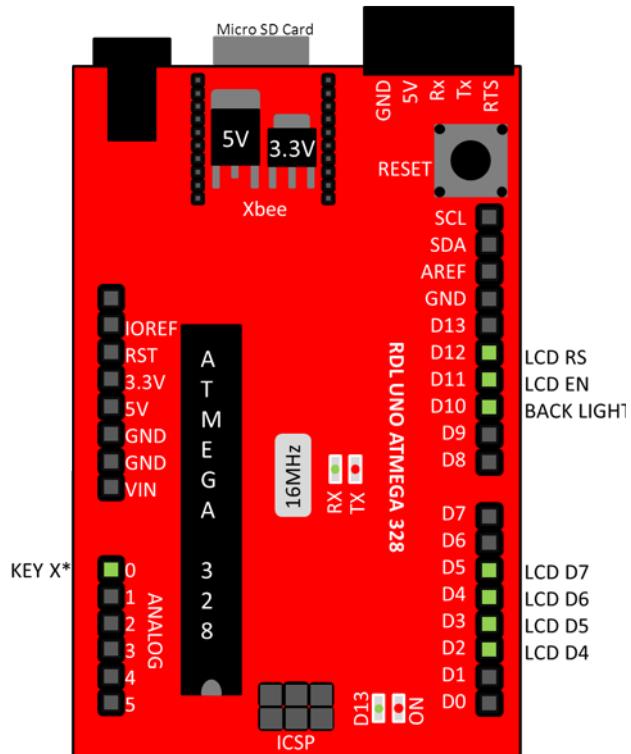


Application interface

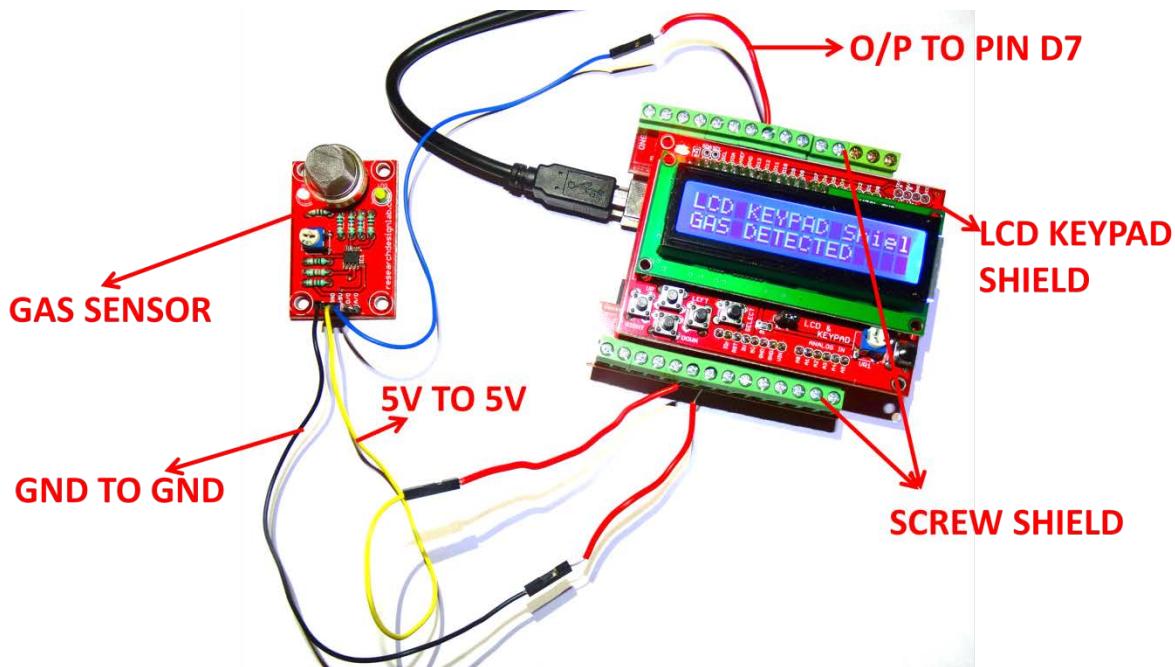
LCD-SCREW SHIELD

OBJECTIVE OF THE EXPERIMENT

If Gas is detected by sensor , "gas detected" message will be displayed in LCD else "gas not detected" message will be displayed in LCD.



*Key X is enabled by placing a jumper on 'KEYS' tag mentioned on the PCB



(Note: for testing, Get a cigarette lighter and half press the lighter button to spill out the GAS.)

LCD AND KEYPAD-SCREW SHIELD ARDUINO CODE

```
/*
* Project name:
LCD KEYPAD Shield
* Copyright
(c) Researchdesignlab.com
* Description:
If Gas is detected by sensor , "gas detected" message will be displayed in LCD
Else "gas not detected" message will be displayed in LCD.
```

The circuit:

- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * 10K resistor:
* ends to +5V and ground



```
* wiper to LCD VO pin (pin 3)
*/
#include <LiquidCrystal.h>      // include the library code:
int sensorValue = 0;           // value read from the keypad
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // initialize the library with the numbers of the interface pins
int sensorPin = A0;

void setup()
{
lcd.begin(16, 2);            // set up the LCD's number of columns and rows:
pinMode(7, INPUT );
delay(2000);
}

void loop() {
lcd.clear();                  // clear lcd display
lcd.setCursor(0, 0);          // set the cursor to column 0, line 0
lcd.print("LCD KEYPAD Shield");
lcd.setCursor(0, 1);          // set the cursor to column 0, line 1

if(digitalRead(7) == HIGH)
lcd.print("GAS DETECTED");
else
lcd.print("GAS NOT DETECTED");
delay(500);
}
```

XBee interface

- Mount the XBee as shown in the pic
- connect the ATmega328 shield to the pc using FT232 connector



Code

```

void setup()
{
    // Open serial communications and wait for port to open:
    Serial.begin(9600);
    while (!Serial) {
        ; // wait for serial port to connect.
    }

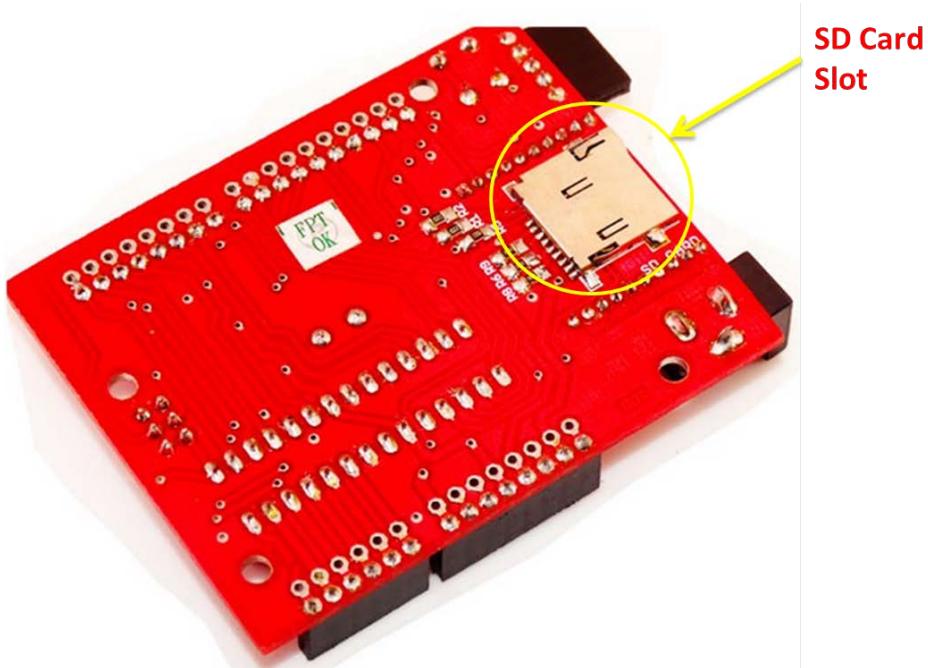
    Serial.println("hellos!");
}

```

{

```
void loop() // run over and over
{
    if (Serial.available())
        Serial.write(Serial.read());
}
```

SD Card Interface



Code

```
#include <SD.h>

File myFile;

void setup()
{
    // Open serial communications and wait for port to open:
    Serial.begin(9600);
    while (!Serial) {
        ; // wait for serial port to connect.
```



{

```
Serial.print("Initializing SD card...");  
// On the Ethernet Shield, CS is pin 4. It's set as an output by default.  
// Note that even if it's not used as the CS pin, the hardware SS pin  
// (10 on most Arduino boards, 53 on the Mega) must be left as an output  
// or the SD library functions will not work.  
pinMode(10, OUTPUT);  
  
if (!SD.begin(4)) {  
    Serial.println("initialization failed!");  
    return;  
}  
Serial.println("initialization done.");  
  
// open the file. note that only one file can be open at a time,  
// so you have to close this one before opening another.  
myFile = SD.open("test.txt", FILE_WRITE);  
  
// if the file opened okay, write to it:  
if (myFile) {  
    Serial.print("Writing to test.txt...");  
    myFile.println("testing 1, 2, 3.");  
    // close the file:  
    myFile.close();  
    Serial.println("done.");  
} else {  
    // if the file didn't open, print an error:  
    Serial.println("error opening test.txt");  
}  
  
// re-open the file for reading:  
myFile = SD.open("test.txt");  
if (myFile) {  
    Serial.println("test.txt:");  
  
    // read from the file until there's nothing else in it:  
    while (myFile.available()) {  
        Serial.write(myFile.read());  
    }  
    // close the file:  
    myFile.close();  
} else {  
    // if the file didn't open, print an error:  
    Serial.println("error opening test.txt");
```

```
}
```

```
}
```

```
void loop()
```

```
{
```

```
    // nothing happens after setup
```

```
}
```

ARDUINO EXPERIMENT BOOK

<http://researchdesignlab.com/projects/Arduino%20experiments.pdf>

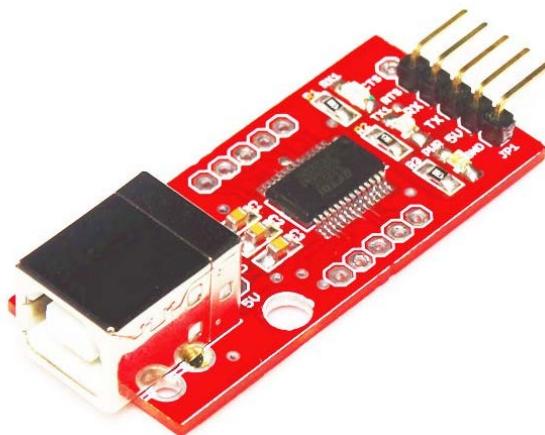
ARDUINO IDE

<http://researchdesignlab.com/projects/UNO%20MANUAL.pdf>



Related Products

FT232 Breakout Board



XBee



GSM GPRS SHIELD



LCD KEYPAD SHIELD

