Study Notes of Iteaduino Part II

- Digital Pin to Control LED Lamp

Digital pin on Iteaduino board can input both high and low electrical levels, which can be used in digital communication or to control external devices. In this chapter, I will briefly describe output of digital pin on Iteaduino, and I will use Iteaduino to control ON/OFF of an LED lamp as a test.

To do this test, we need:

- Iteaduino board x 1
- LED electronic brick x 1

Construction of hardware circuit

D0-D13 pins on Iteaduino are all digital pins, and we choose D9 pin to control ON/OFF of the LED lamp. All of the electronic brick modules adopt unified 3PIN interface for easy connection to Iteaduino board. We only need to connect the LED electronic brick module to D9 pin on Iteaduino with the connection cable (as shown in Figure 1).

After the above connection, cathode of the LED lamp is linked to the GND terminal on Iteaduino board, and anode is linked to D9 pin on Iteaduino via S pin. When D9 is controlled to output high level, LED lamp will be ON; while when D9 is controlled to output low level, LED lamp will be OFF.

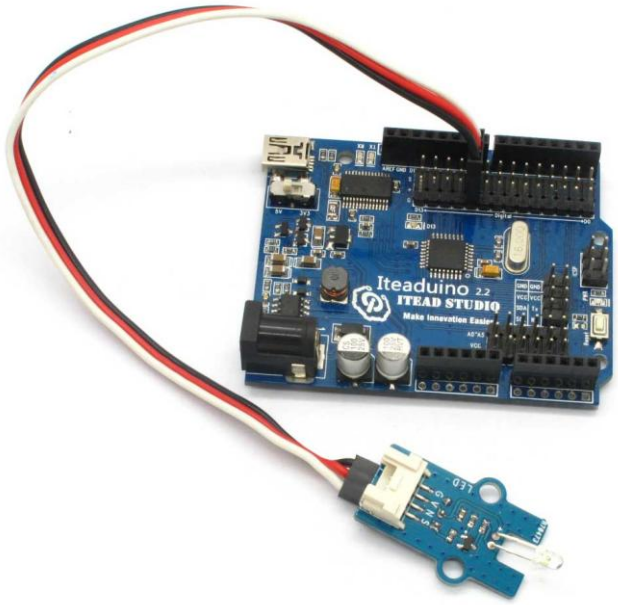

Figure 1

Writing of control program

First, I’ d like a brief introduction to Arduino program structure:

Setup (): It is an initialization function used to assign an initial value to a variable or to configure the pin mode, etc. During this test, the setup () function part will configure the pins used under an output state.

Loop (): This is the main loop function of Arduino program, which equals to Main () function. During this test, we use the codes in loop to control ON/OFF of LED lamp.

Next, I will briefly describe Arduino library functions used in the test：

pinMode(pin, mode)

Function：to configure the specified pin as output or input.

Parameters：

- Pin: to set pin N.O., such as “0” for D0 and “7” for D7.

- Mode: to set mode, INPUT or OUTPUT.

For example, pinMode(2, output) means to set D2 pin under output state.

（Note:It is difficult to use D13 pin as a digital input, as most of the control boards use an LED and a resistor to connect it. If an internal 20K pull-up resistor is started, its voltage will be around 1.7V instead of the normal 5V, as the onboard LED series resistor pulls it down, which means that its returned value is always LOW . If input mode of D13 digital pin must be used, an external pull-down resistor is needed）

digitalWrite(pin, value)

Function： to control the interface to output high level or to input low level

Parameters：

- Pin: pin N.O. for assignment, such as “0” for D0 and “7” for D7.
- Value: output level, can be high or low.

For example, digitalWrite(2, high) means to control D2 to output high level

The complete program used in the test is as below:

```
int ledPin=9; //to set the digital IO pins to control LED
void setup()
{
    pinMode(ledPin,OUTPUT);//to set mode of the digital IO interface
}
void loop()
{
    digitalWrite(ledPin,HIGH); //to set PIN5 as HIGH, about 5V
```

```
    delay(2000); //to set delay time , 2000 = 2s
    digitalWrite(ledPin,LOW); //to set PIN5 as LOW, equals to 0V
    delay(2000); //to set delay time , 2000 = 2s
}
```

Compiling and uploading of program

Next, we need to compile the above program and upload it to ATMega328 master control in Iteaduino board. The steps are as follows:

Firstly, : open Arduino IDE, copy the above control program and paste it into Arduino IDE, as shown in figure 2.

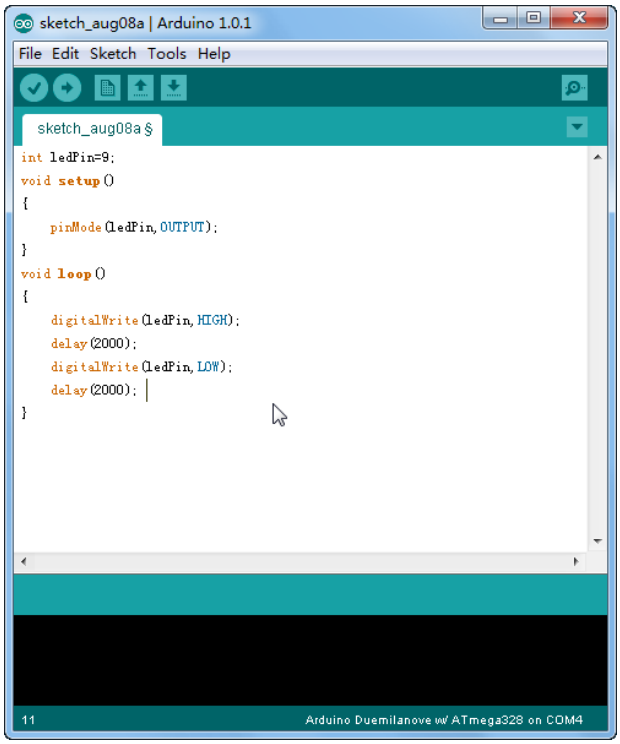

Figure 2

Secondly, select the model corresponding to Iteaduino board which you are using under Arduino IDE menu “Tools” > “Board” . For Iteaduino 2.2 , we should choose "**Arduino Duemilanove or Nano w/**

ATmega328", as shown in figure 3. If you use the Iteaduino UNO but not the Iteaduino board, you need to choose the "Arduino UNO" selection instead of using the Duemilanove.

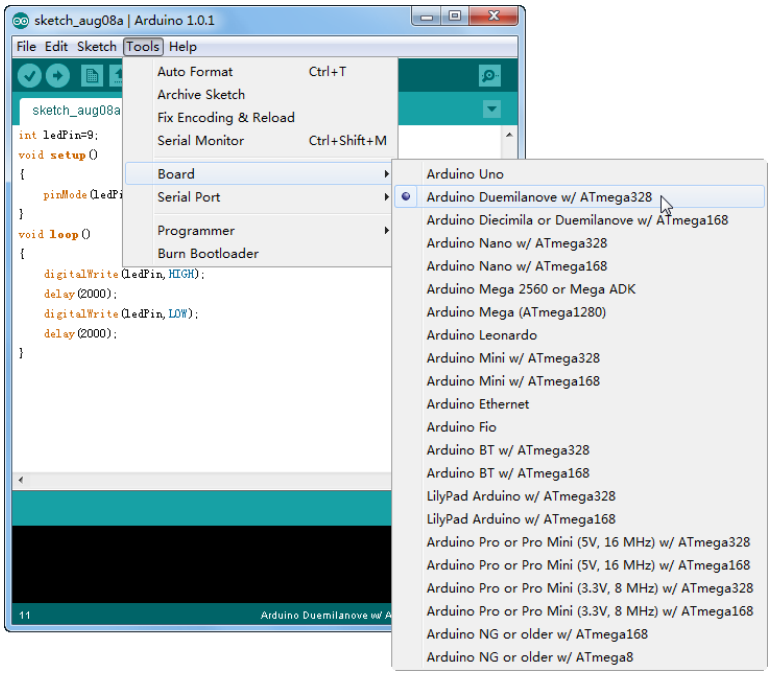

Figure 3

Thirdly：select serial COM number. If you are not sure about the serial number of board, you can check it under the device manager of computer. As shown in figure 4, we can confirm that the serial number of Iteaduino in the computer is COM15.

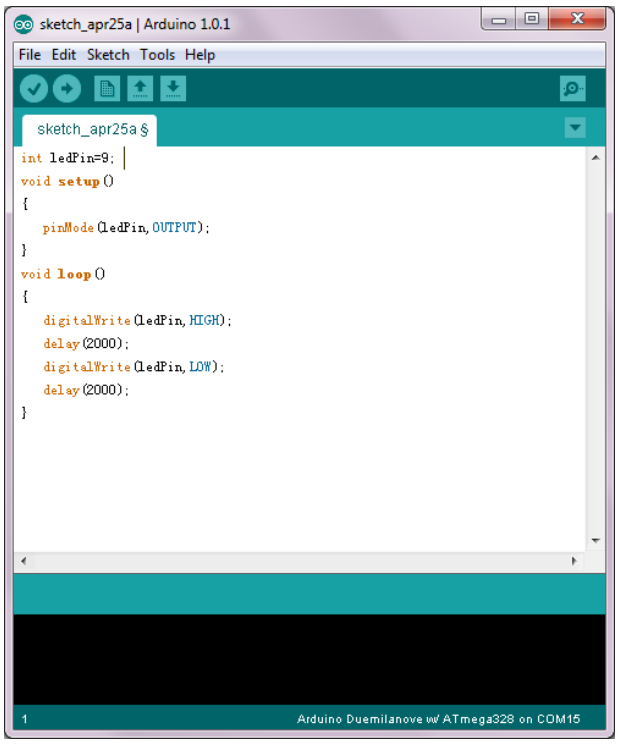

Figure 3

After serial number is confirmed, check the corresponding serial number under menu **Tools | Serial Port**, as shown in figure 5.

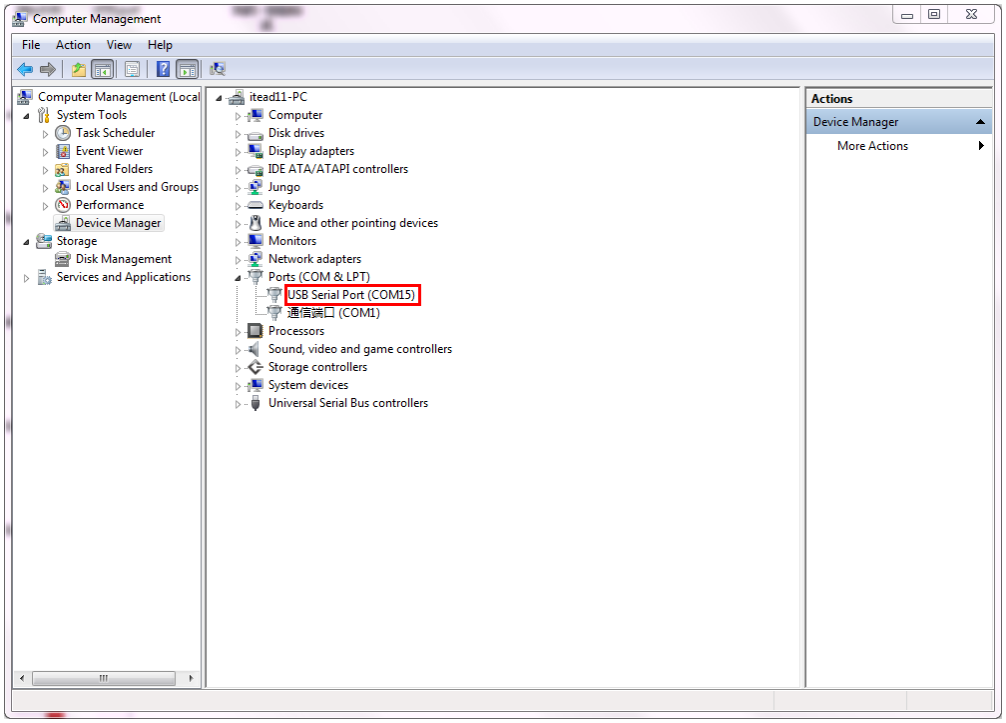

Figure 5

Fourthly, click the compiling button to compile the program (as shown in figure 6), you can check if there are errors in the program through compiling.

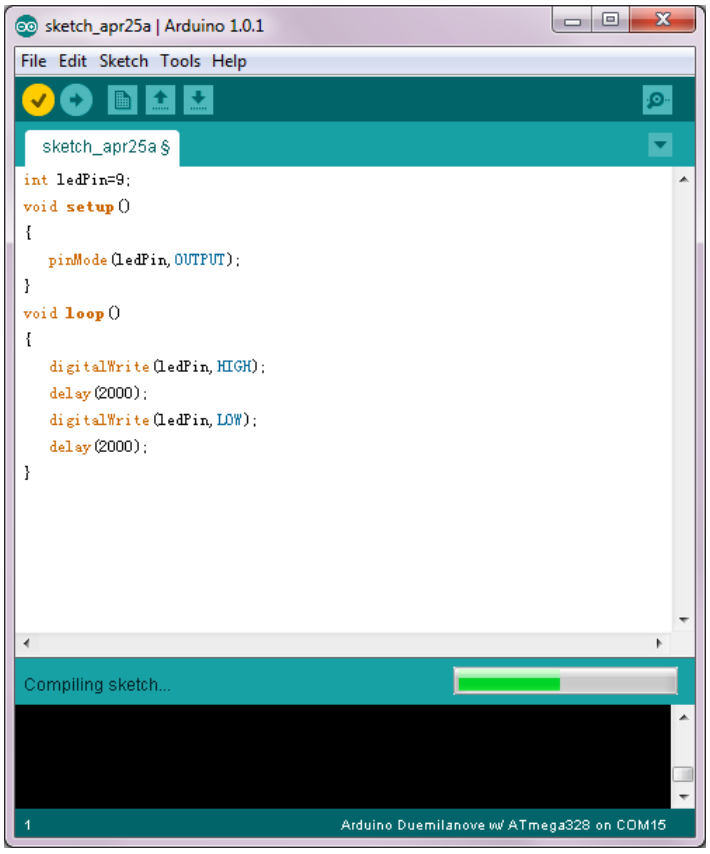

Figure 6

If compiling failed due to errors in the program, errors will be indicated in the prompt window. If compiling succeeded, the file size after compiling will be shown in the prompt window. As shown in figure 7, the compiled program size is 1084 bytes.

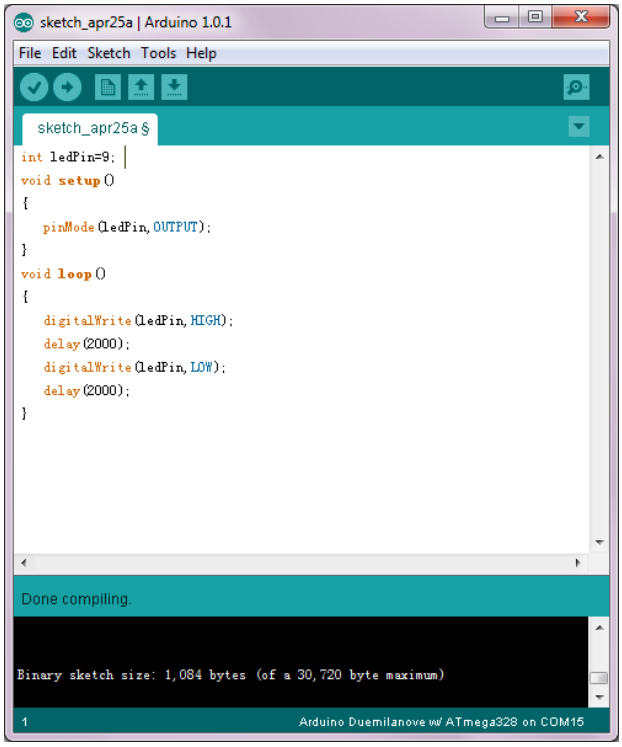

Figure 7

Finally, click uploading button (as shown in figure 8), the program is rewritten into ATMega 328 chip on Iteaduino after compiling.

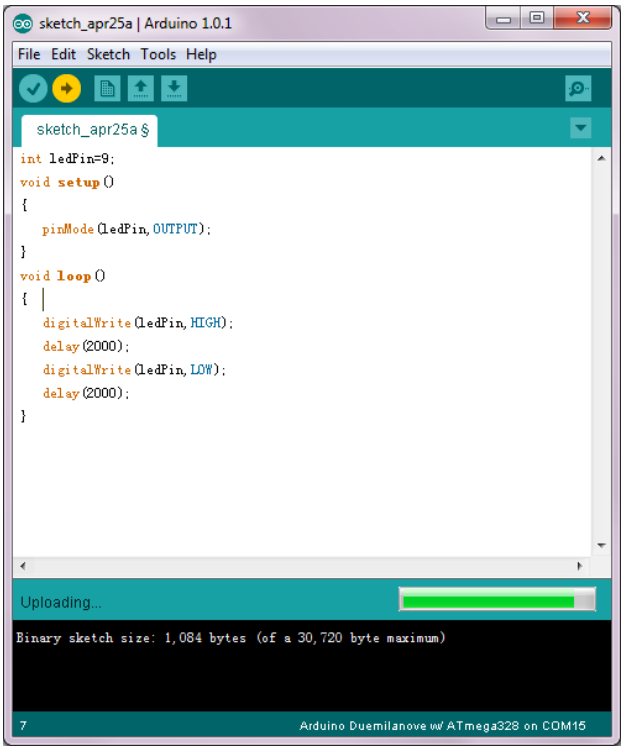

Figure 8

If uploading succeeded, "Done uploading" will be prompted, as shown in figure 9,

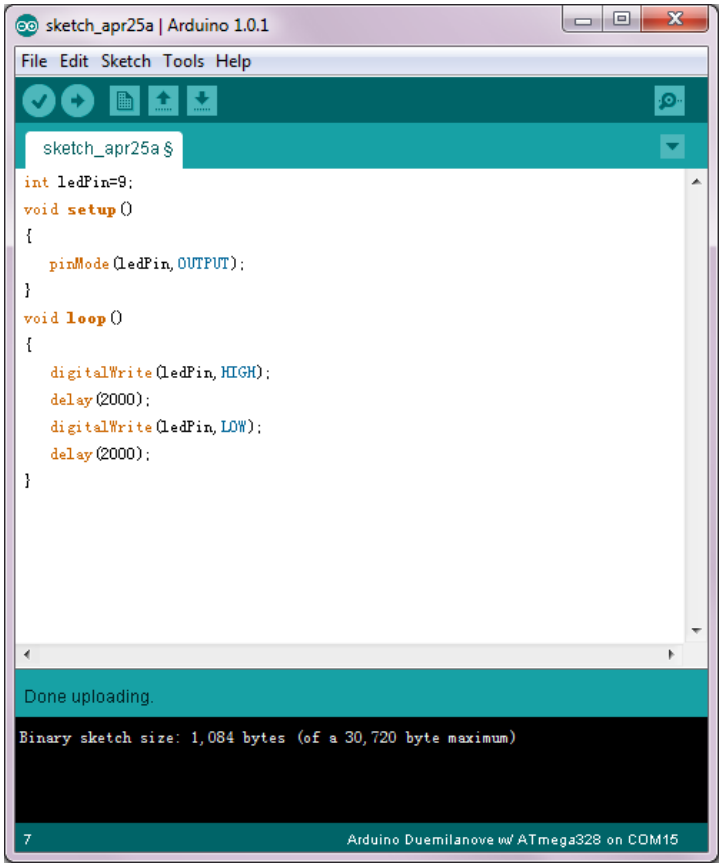

Figure 9

After successfully downloading program to Iteaduino board, the LED lamp in the electronic brick will keep flashing.

In the test above, we have done the test of controlling ON/OFF of LED lamp by D9 pin on Iteaduino. Now I' d like to do another test to control the brightness of the LED lamp changing from dark to bright through D9 pin on Iteaduino.

The brightness of the LED lamp is related to the voltage at the two terminals, and within the normal range, the higher the voltage, the brighter the LED lamp. Therefore, we can adjust the voltage at two terminals of the LED lamp by Iteaduino to achieve control on brightness.

Among the digital IO interfaces D0-D13, D3, D5, D6, D9, D10 and D11 support PWM output . PWM, the abbreviation for Pulse Width Modulation, is a method of digitally encoding analog signal levels, which is to encode the electrical level of a specific analog signal by changing the duty cycle. Due to the ordinary digital I / O port output of Iteaduino, only high level(5V or 3.3V) or low level(0V) can be input, and no continuous analog voltage value can be generated. Therefore, pulse width modulation technology must be adopted to simulate a voltage between 0 and 5V.

Construction of hardware circuit

Hardware connection is the same with that in above test, we only need a different control program.

Writing of control program

Here we will use a new Arduino library function:

analogWrite(pin, value)

Function：to output an equivalent analog voltage via PWM control

Parameters：

- Pin: Pin N.O. for assignment, such as “0” for D0 and “7” for D7.
- Value: Range is 0~255. Duty cycle=value/256, actual output a waveform with a certain duty cycle and 490Hz frequency .

For example, analogWrite(9,64) means D9 will output a rectangular wave with duty cycle of 25% and 490Hz frequency.

The complete program used in the test is as below:

```
int brightness = 0;      //To define the integer variable (brightness) and its initial value, the variable indicates the brightness of LED lamp.
int fadeAmount = 5;      //To define the integer variable (fadeAmount) ,the variable indicates the amount of brightness change.
void setup()   {
  pinMode(9, OUTPUT);// To set N.O.9 interface as output interface:
}
void loop()   {
  analogWrite(9, brightness);//To write the brightness value into N.O.9 interface
  brightness = brightness + fadeAmount;//To change the brightness value so that the brightness will be changed in the next cycle.
  if (brightness == 0 || brightness == 255) {
     fadeAmount = -fadeAmount ; //turn over when brightness reaches the highest and lowest point
  }
  delay(100); //Delay by 100ms
}
```

Then compile and download the above program to Iteaduino. As the operation is the same with that in last test, I will not repeat it here.

After the program is downloaded, you will find that LED lamp in the electronic brick module turns from dark to bright and then from bright to dark and the cycle repeats continuously.