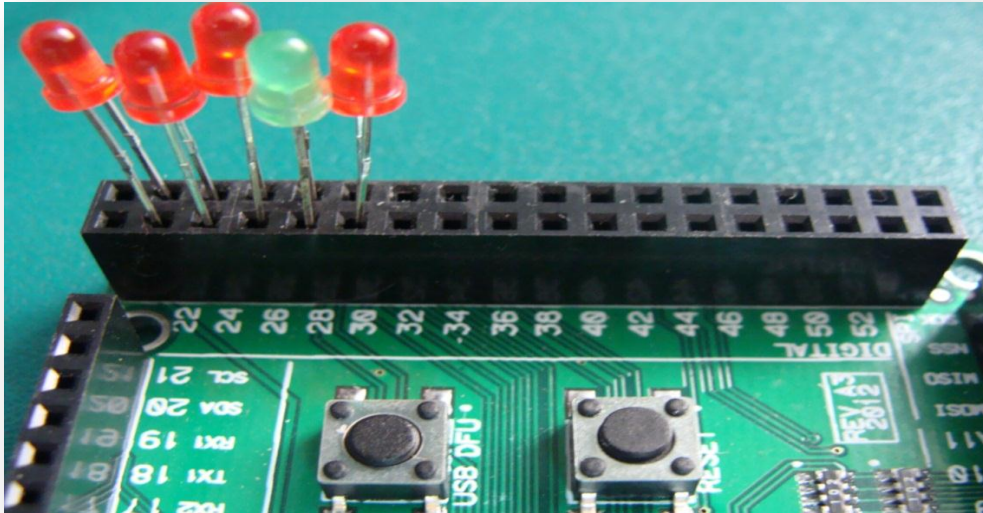


## 初级例程之一跑马灯

程序比较简单，是以 ART\_WORK 中的例程 Blink 修改而来  
硬件连接如下图



利用数字 IO 口 22—31，偶数端口连接正极，奇数端口连接负极，共 5 个发光二极管  
代码也很简单，首先在 setup 中对端口做初始化，art 是基于 arduino 发展而来，对于端口的操作采用每个 IO 单独操作，这里利用数组和循环的方式可以简化操作

```
int led[5] = {22, 24, 26, 28, 30};  
int gnd[5] = {23, 25, 27, 29, 31};
```

定义 2 个数组，分别用来保存偶数 Pin 和奇数 Pin，

```
void setup() {  
  // initialize the digital pin as an output.  
  for (int i=0; i<5; i++)  
  {  
    pinMode(led[i], OUTPUT);  
    pinMode(gnd[i], OUTPUT);  
    digitalWrite(gnd[i], LOW);  
  }  
}
```

先将 IO 初始化为输出模式，然后将所有的 gnd 置 0，借助 STM32 的推挽输出我们只要修改 led IO 的高低电平就可以驱动 LED。如果是 51 芯片的话这种方式就不行了

初始化完成之后，我们就可以对 LED 操作了，修改下面的代码，就可以编出各种花样的跑马灯

论坛: <http://www.rt-thread.org/phpBB3/viewforum.php?f=27>

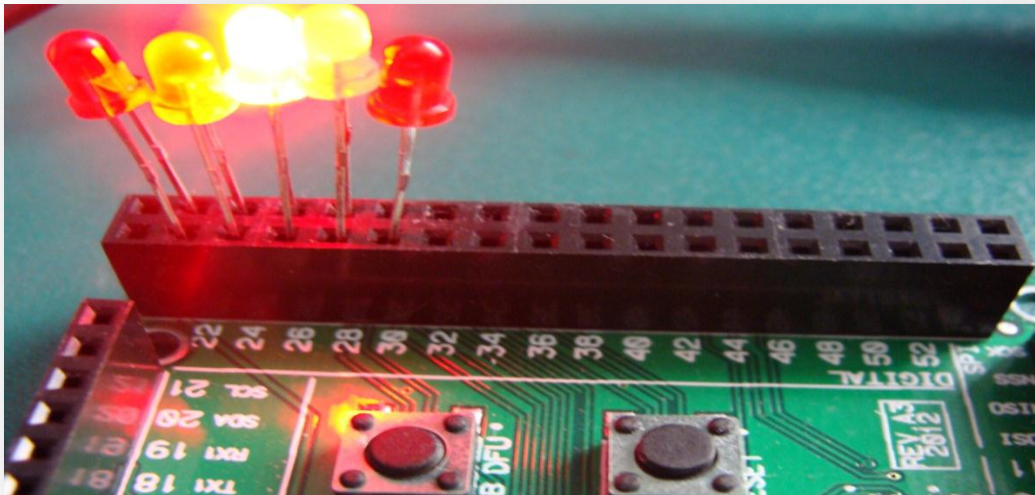
相关文档：

<http://pan.baidu.com/netdisk/extractpublic?uk=4264488348#dir/path=2551034%3A%2FART%E6%96%87%E6%A1%A3>

```
int i = 0;
// the loop routine runs over and over again forever:
void loop() {

  digitalWrite(led[i], HIGH); // turn the LED on HIGH
  delay(1000);                // wait for a second
  digitalWrite(led[i], LOW);  // turn the LED off by
  delay(1000);                // wait for a second
  if(i++>=5)
  {
    i=0;
  }
}
```

下图是运行效果，照片上无法看到动态效果



扩展阅读：

熟悉单片机编程而没有接触过 arduino 的，可能会有疑问，为何没有看到 main 函数，其实 main 同样存在，只不过为了简化而隐藏了，在 main.cpp 中可以找到 main 函数，以及对 setup 和 loop 的调用

```
int __main(void)
{
  construct();
  init();
  setup();
  for (;;) {
    loop();
  }
  return 0;
}
```

同样遵循易用原则，在 art 中对端口初始化只要执行 pinmode 就可以，但如果用固件库来编程就复杂了，以 pinmode 的 output 实现为例，如下图，初始化一个 IO,需要先打开

论坛：<http://www.rt-thread.org/phpBB3/viewforum.php?f=27>

相关文档：

<http://pan.baidu.com/netdisk/extractpublic?uk=4264488348#dir/path=2551034%3A%2FART%E6%96%87%E6%A1%A3>

```
void pinMode(uint8_t pin, uint8_t mode)
{
    const struct pin_index* index;
    if (pin < ITEM_NUM(pins))
    {
        GPIO_InitTypeDef GPIO_InitStructure;
        index = pins[pin];

        /* GPIO Periph clock enable */
        RCC_AHB1PeriphClockCmd(index->rcc, ENABLE);

        /* Configure PE5 */
        GPIO_InitStructure.GPIO_Pin = index->pin;
        if (mode == OUTPUT)
        {
            /* output setting */
            GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
            GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
            GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
            GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
        }
    }
}
```

时钟，然后配置 IO 模式，输出类型，端口速度，上拉或下拉，然后才能初始化结束

同样，又有点就会有缺点，易用的同时降低了灵活性，比如 art 中一次只能对一个 IO 初始化，并且只能遵循 pinmode 中对 output 的实现，输出只能设置为推挽模式。传统方式一次可以对多个 IO 初始化，并且可以对 IO 的输出模式可以做更多的设置

论坛: <http://www.rt-thread.org/phpBB3/viewforum.php?f=27>

相关文档：

<http://pan.baidu.com/netdisk/extractpublic?uk=4264488348#dir/path=2551034%3A%2FART%E6%96%87%E6%A1%A3>