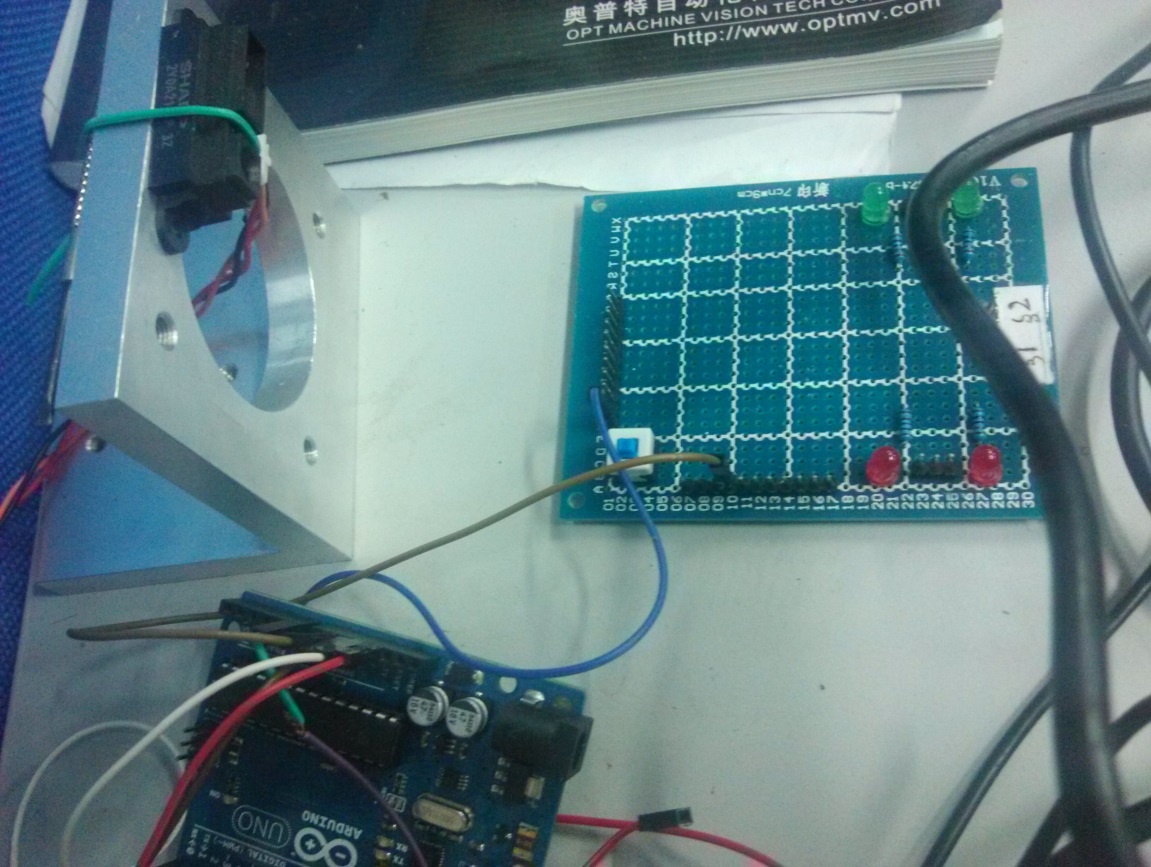
**基于Arduino的PID测距控制**

Kevinzhang

Arduino作为现在最火的开源硬件平台之一，已经在世界上掀起众多电子爱好者学习的浪潮。笔者作为学生，初了拿它实现简单的IO口控制外，还想加入更多的元素，这次计划加入一个PID控制算法来实现位置跟随运动，以求追求更加高大上的应用体验。

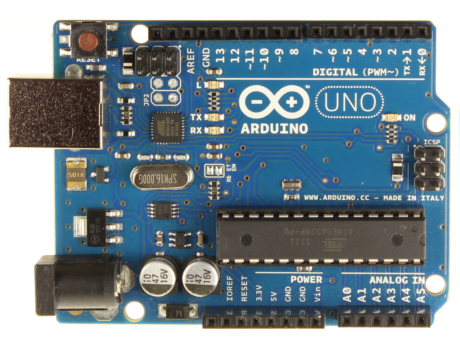
实验初步硬件平台如下：



位置检测元件使用的是一个夏普红外传感器，其技术规格如下所示

* 信号类型：模拟输出
* 探测距离：4-30cm
* 工作电压：4.5～5.5V
* 标准电流：33mA
* 接口类型：PH2.0-3P
* 最大尺寸：40x20x13.5 mm

由于位置传感器输出的是模拟量，需要在通过Arduino接收后再转化为数字量，使用的Arduino UNO如下图所示，



传感器的输出量的参考转换代码如下

void loop ()

{

uint16\_t value = analogRead (pin);

if (value < 16)

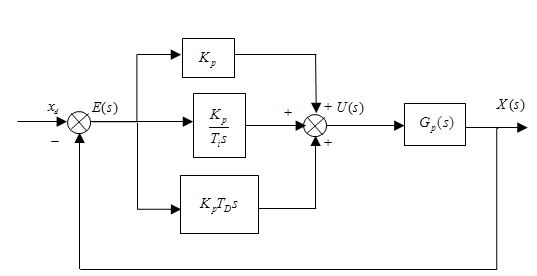
value = 16;

distance= 2076.0 / (value - 11.0);}

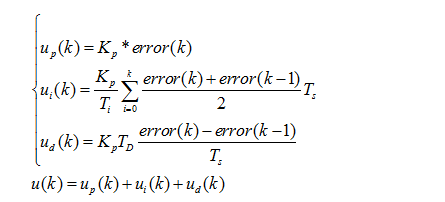
其中pin为模拟量输入引脚号（A0-A5），distance为转换成的数字量。

**PID算法**

也许学过控制课程的童鞋都知道PID算法，它有什么作用呢？作为改变世界的十大算法之一，当然有必要自己尝试一下咯，也许只有你自己接触后，才会有更深的体会。笔者认为，PID算法能够使闭环控制系统能够更快更好的实现我们的预期功能，先介绍下我们的算法原型：



上面所示的是PID控制原理框图，我们需要将其在程序中离散化

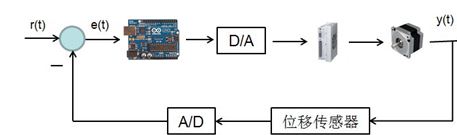


这样看似高深的东西可以简化为每次采样完成后的计算输出，我们计划采用u作为控制器的控制量，控制电机的转动，电机的输出速度与输入电压成正比例关系，拟采用电机转动的位置来模拟最终的调节量，如果传感器检测到目标位置，电机将停止转动。

# 基于Arduino的PID测距控制（二）

在上期的blogger中，笔者讲述了如何使用Arduino和夏普红外传感器来检测距离。并就PID的基本理论进行阐述。在本期blogger中，笔者将以电机的控制电压量变化演示如何实现PID反馈闭环控制。

下图是控制原理框图



由于不知道控制对象具体的理论传递函数，我们在进行PID参数的选取时进行了若干次的尝试，直到相对满意时为止。

**如何定义一个ms采样周期？**

Arduino的millis()函数能够获取arduino从开始运行到现在的程序之后的毫秒数，返回值为unsignd long型。通过设置一个timechang=now-lasttime，阈值设为100时，即表示100ms的时间间隔。

**定义采样周期计算Arduino的输出值**

那么如何将PID计算输出呢？

   up=Kp\*error;

   ud=Kd\*(error-error\_1)/Ts;

   ui=ui\_1+Ki\*error\*Ts;

输出u=up+ud+ui，Ts表示采样周期，error为当前检测到距离距离目标位移的差值，而error\_1为上一次采样的差值，这样既可以将上期bloggger中的离散控制算法转换为C语言。

**监控Arduino的控制量**

通过Serial.print ()，通过Arduino的Serial Monitor显示出来，附件为参考的代码。

