Developers Diary for propeller powered Levitron:

Reinhard Mayer **January 2012**

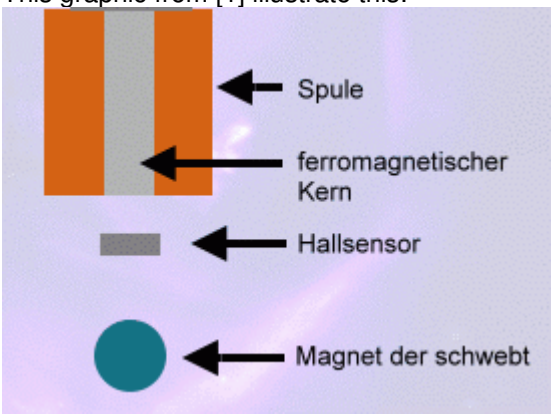
The basic idea and mother of all was coined from here: [1]
http://www.bis0uhr.de/index.htm?http...dex.html%99htt
In opposite to the analog design [1] I have the goal to do this with a microcontroller.

The rest is very similar to [1].

I have a software generated PWM signal in a range from 0% to 100%.
This signal controlls a Power Transistor, 0% means no current flows through the transistor, 100% means the maximal current flows.

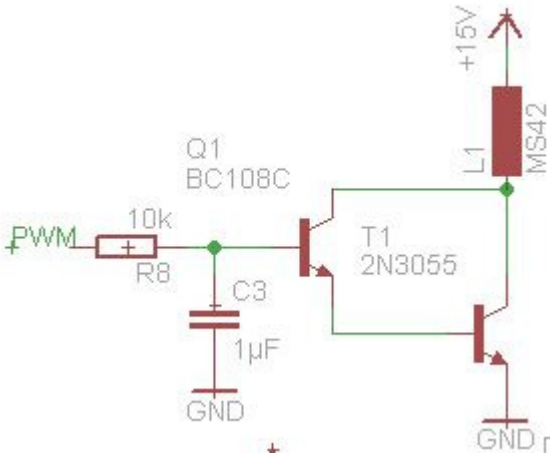
This current flows also through the solenoid and produce a magnetic field, about squarish proportional to the current.
The field flow through a magnetic sensor (hall sensor), which is in circa 2cm distance from the solenoid, generate a force
on a magnetic object behind the sensor.

This graphic from [1] illustrate this:



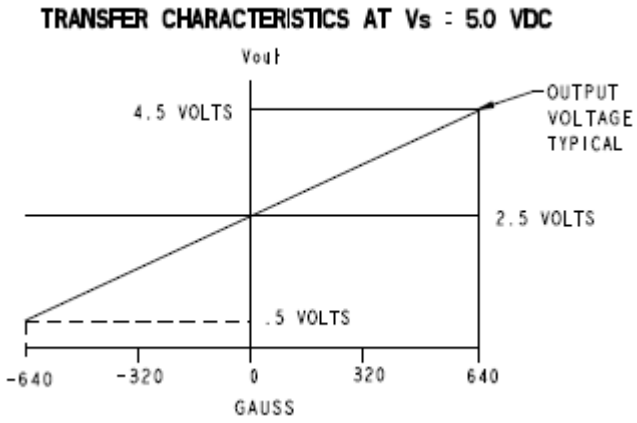
Spule ... solenoid
ferromagnetic core
hallsensor
floating magnetic object

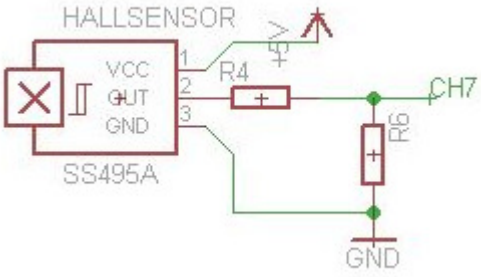
So now we have the signal path: PWM - Transistor - Solenoid
this is shown here:



Now we measure the field flow through the hallsensor.

This sensor outputs a voltage with the slope 3.125 mV/Gauss.
This means if no field around, the sensor outputs 2.500 V (at 5 Volt Power Supply)





This output voltage is digitize to a 12bit digital value
and this value is read by the microcontroller software.

```
x = readMAX147(0xFF);   // read actual value
```

the readMAX147 function is part of my common used Library
the parameter 0xFF means : read Channel7, unipolar, with external clock.

Now is the question how can we get the value for the desired position of the object.
I tried it with this approach.
Set the PWM to 50%, half time ON other half time OFF
Bring the object (ball) in desired position.
Read the value at this point under this condition.

```
void getRefPos()
{
    short x;
    DIRA |= 0xFF;
    PWM = 50;
    while(1)
    {
        x = readMAX147(0xFF);
        OUTA = x >> 2;
        printf("%d\n",x);
    }
}
```

So I get a fairly stable value, called it w.

w is the set point for every closed loop control.

Now we can do the simplest control of all: a 2 level controller or comparator

```
    while(1)
    {
        x = readMAX147(0xFF);   // read actual value

        OUTA = x >> 2;          // scale to 8 bit for DAC

        if(x < w)
            PWM = 80;
        else
            PWM = 0;

    }
```

Note in opposite to the analog design, we can switch between two points,
what not must be categorical 0% and 100%.
Play with this 2 Points gets a lot.

best results are arrived with a PID control loop.
With it are stable states for hours possible.

```
void pid_method ()
{
    short w,x,e,e2,ei,y;

    #define Kp3 40
    #define Kd2  25
    #define Ki 4

    w = W;
    e2 = 0;
    ei = 0;
    DIRA |= 0xFF;

    while(1)
    {
        x = readMAX147(0xFF);   // read actual value

        OUTA = x >> 2;          // scale to 8 bit for DAC

        e = w - x;              // error

        ei += e;

        y = e * Kp3 + (e - e2) * Kd2  + ei * Ki;   // control output

        e2 = e;

        PWM = y ;


    }
}
```