MechatronicDesignCases 心得

11.2 Velocity Control of the dc Motor Kit

This motor is very important since it comes with a gearbox (ratio 6:1) and
an encoder that gives one hundred pulses per revolution, which gives 600 pulses per
revolution that we exploit by using quadrature method to bring it to two thousand
four hundred pulses per revolution.

這個馬達非常重要因它有一變速箱 6:1 跟一編碼器可給一百脈衝每轉，每轉產
生 600 個脈衝，我們使用正交方法加以利用，使其達到每轉 2440 個脈衝。

脈衝(pulses): 訊號振幅的快速暫態變化，由基準值變為較高或較低的值，之後
又快速的回到基準值。

它這裡好像沒有提到脈衝對馬達每轉的影響。

程式碼:

```
#include "p30F4011.h" // 將名為 p30F4011.h 的程式加入進來
// 定義一個  struct
typedef struct {
float K_P; // Propotional gain
float K_I; // Integral gain
float Const1_pid; // KP + T KI/2
float Const2_pid; // -KP + T KI/2
float Reference; // speed reference
float y_k; // y_m[k] -> measured output at time k
float u_k; // u[k] -> output at time k
float e_k; // e[k] -> error at time k
float u_prec; // u[k-1] -> output at time k-1
float e_prec; // e[k-1] -> error at time k-1
}PIStruct;
PIStruct thePI;
thePI.Const1= thePI.K_P+T*thePI.K_I/2;
thePI.Const2=-thePI.K_P+T*thePI.K_I/2;
thePI.Reference=600;
float ReadSpeed(void);
float ComputeControl(void);
float SendControl(void);
void __ISR _T1Interrupt(void) // interrupt routine code
{
float Position_error;
```

```
thePI.y_m=ReadSpeed();
thePI.e_k= thePI.Reference-thePI.y_m;
ComputeContrl();

SendControl();
IFS0bits.T1IF=0; // Disable the interrupt
}
int main ( void ) // start of main application code
{
int i;
from inputs) Reference = 0x8000; // Hexadecimal number
(0b... Binary number) ThePID = 0x8000;
TRISC=0x9fff; // RC13 and RC14 (pins 15 and 16) are configured as outputs
IEC0bits.T1IE=1; // Enable the interrupt on Timer 1
while (1)
{
}
return 0
}
% ReadSpeed function
int ReadSpeed (void)
{
}
% ComputeControl function
int ComputeControl (void)
{
thePI.u_k=thePI.u_prec+thePI.Const1*thePI.e_k+thePI.Const2*thePI.e_prec;
}
% SendControl function
int Send Control (void)
{
sendControl()
thePI.u_prec=thePI.u_k;
ThePI.e_prec=thePI.e_k;
}
```

上述這些程式碼都很難看懂，書上也沒有說用在哪裡。