

周期誤差補正装置の開発

2008年8月12日 市川隆 (東北大理)

1. はじめに

開発した AIR-T-40 (40cm 南極赤外線望遠鏡)には 4 分周期の追尾誤差がある(図 1)。これは望遠鏡の追尾に使われているウォームギアの偏芯に起因するものと思われる。その大きさは望遠鏡のバランスを十分に良く調整したあと、peak-to-peak で約±2"である。40cm 望遠鏡の 0.5 μm 波長での回折限界が 0.26"であることを考えると、シーイングの良いことが期待されるドームふじにおいては無視できない追尾誤差である。ウォームギアの原点検出と望遠鏡制御ソフトの改良で対応は可能であるが、制御ソフトは特注とは言え市販のものなので、改良は困難である。幸い、制御システムにはガイド用信号の入力ポートがあるので、そのポートに周期的に補正信号を送るハードウェアを開発することにした。

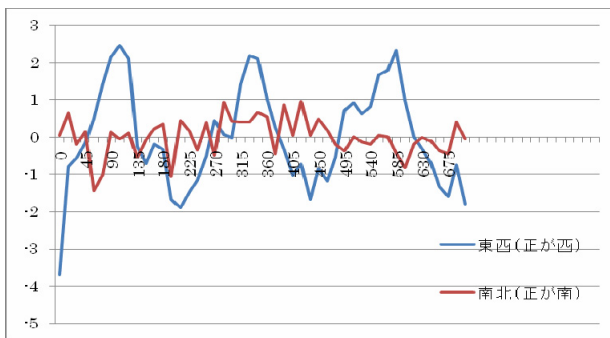


図 1 周期誤差の例(2008年2月15日)

2. マイコンによる制御システム

Texas Instruments 社のミックスド・シグナル・マイコンプロセッサ MSP430F4270 は、16 ビットタイマ、16 ビット A/D コンバータ、12 ビット D/A コンバータ、32 個の I/O 端子、LCD ディスプレイ・ドライバを搭載した 16 ビット汎用マイコンである(トランジスタ技術 2007年1月号に特集あり)。しかも 1 個 1200 円と安価である。4K バントまでのプログラムならば、無償で開発プラットフォームが使える C 言語で開発ができる。

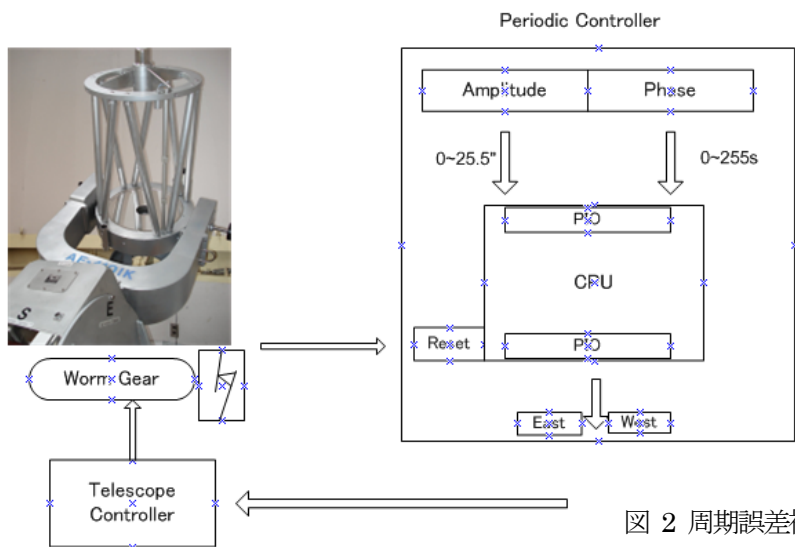
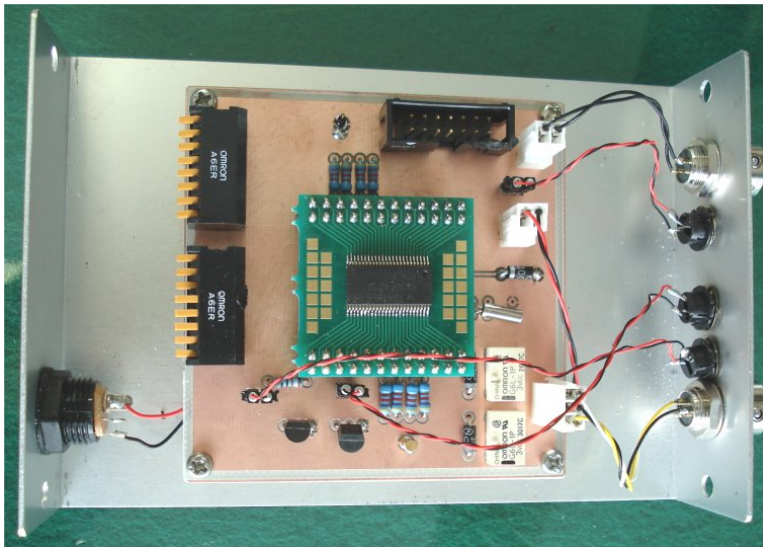
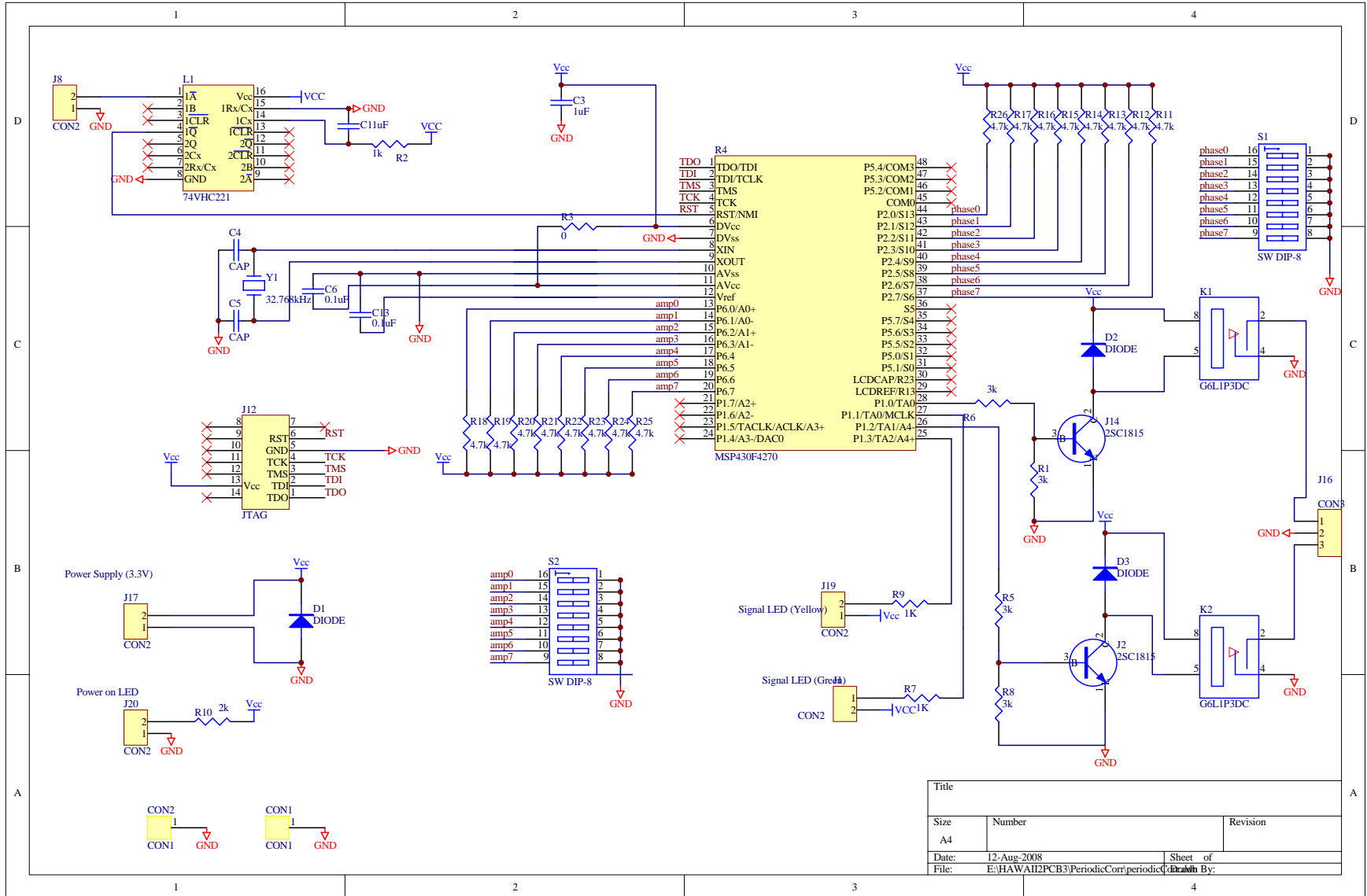


図 2 周期誤差補正システム

図 2 に周期誤差補正システムの構成を示す。周期誤差の振幅は 8 ビット DIP スイッチで 0.1" 単位の 2 進数で入力される。ウォームギアと周期誤差の位相の差は、1 秒単位で同様に入力される。CPU には内部のタイマーから 1 秒毎に割り込みがかかり、そのつど、補正のための追尾信号が望遠鏡制御装置に送られる。リレーによって、望遠鏡制御装置のガイド入力ポートの East または West の一方を GND に接続する。その間ステップモーターに駆動パルスが送られる。周期誤差コンサローラはマイクロスイッチからの信号によってリセットされ、初期値から繰り返される。

図 3. 周期誤差補正コントローラ





Title		
Size	Number	Revision
A4		
Date:	12-Aug-2008	Sheet of
File:	E:\HAWAII\PCB3\PeriodicCorr\periodicCorr.dsn	By:

周期誤差用プログラム

```
*****
// MSP430F4270 periodic correction
//
// Using Watch Dog Timer, pulse for ST4 are sent each 1 sec
*****
#include "msp430x42x0.h"
#define PERIOD 240
#define PERIODHALF 120
#define PULSEUNIT 600 // should be determined by expericence
int offset, amp, pulse;
float x;
void main(void)
{
    WDTCTL = WDT_ADLY_1000; // setting of WDT, interval mode for 1s
    // WDTPW(bus word)+WDTTMSEL(inter. Timer mode) +WDTCNTCL(counter
reset)+WDTSSSEL(ACLK)
    // interrupt interval = 32768(ACLK) / 32768(WDTIS) = 1000ms
    P1DIR = 0x0F; // output direction for P1.0, P1.1, P1.3
    P2DIR = 0x00; // input for P2.0-7. phase (0-255s)
    P6DIR = 0x00; // input for P6.0-7 amplitude(P-P) 0.1" unit
    IE1 |= WDTIE; // enable for WDT Timer
    offset = 255-P2IN; // input of phase in units of second from P2 port
    x = 255-P6IN;
    amp = (int)(x*0.1 +0.5); // convert 0.1 unit to 1 unit
    pulse = (int)((float)amp*PULSEUNIT +0.5); // output period to ST4
    __BIS_SR(LPM0_bits + GIE); // get into LPM0 mode
}
// interrupt for WDT
#pragma vector=WDT_VECTOR
__interrupt void WDT (void)
{
    long i,j;
    if(offset < PERIODHALF) {
        P1OUT = 0xF9;
        for(i=0; i< pulse; i++){
            j++;
        }
    }
}
else{
```

```
P1OUT = 0xF6;
for(i=0; i< pulse; i++){
    j++;
}
}
offset++;
if(offset > PERIOD) offset=0;
P1OUT = 0xFA;
}
```