

3 Prilohy

Príloha 1 – Typy montáží



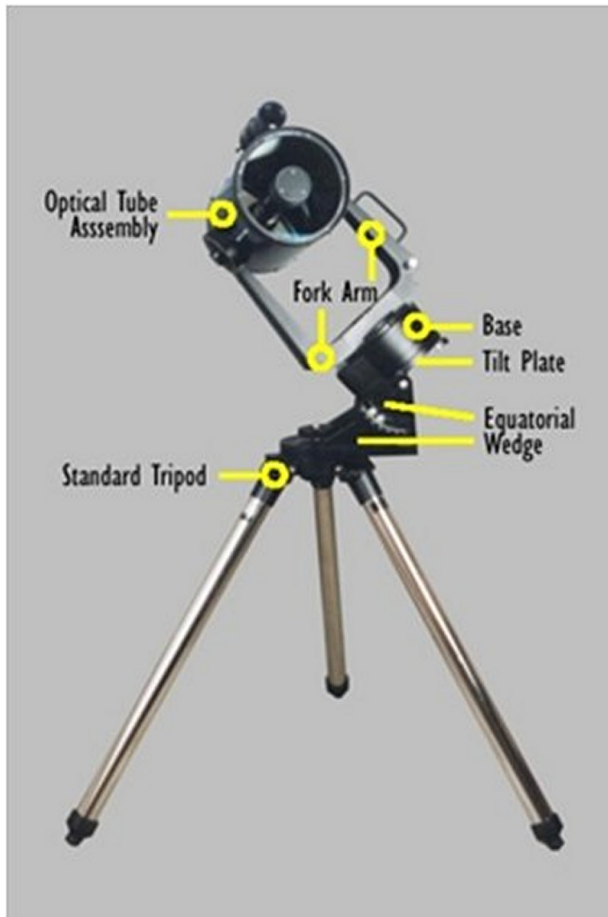
↑ Prvým a zároveň najjednoduchším typom je alt-azimutálna montáž. Na obrázku vyššie je konkrétne typ dobson, ktorý je obľúbený vzhľadom k jednoduchosti stavby aj v domácich podmienkach.



Ďalším typom, je už paralaktická montáž, konkrétne nemecká.

Polárna os je upevnená v zemi alebo na statíve, kolmo na ňu je pripevnená deklinačná os, ktorá má na jednom konci pripevnený teleskop a na druhom konci protizávažie. Jej nevýhodou je skutočnosť, že teleskop nemôže nasledovať objekt po oblohe celý čas, pretože keď dosiahne meridián, tak tuba teleskopu narazí na stĺpik montáže, a teleskop treba pretočiť. V našom projekte bol použitý práve tento typ montáže.

←



Americká, alebo aj vidlicová montáž. Odpadá potreba použitia protizávžia, pretože hmotnosť teleskopu je vyvažovaná hmotnosťou samotnej montáže. Je vhodná pre teleskopy s krátkou tubou (Maksutov-Cassegrain a i.), pretože väčšie by potrebovali dlhšiu vidlicu, čím by sa znížila stabilita celej sústavy. Laik by si povedal, že ide vlastne o naklonenú azimutálnu montáž, pri ktorej sa azimutálna os nahrádza polárnou a výšková deklinačnou.

←



Anglická montáž. Polárna os ne upevnená na obidvoch koncoch. V jej strede je kolmo deklinačná. Vhodné pre teleskopy zo stabilným stanovišťom.

←

Príloha 2 - Princíp fungovania LNB

LNB - „Low-noise Block Converter“ (označovaný aj ako satelitný konvertor) je zariadenie umiestnené v ohnisku parabolickej antény, ktoré prijíma signál z družice (v našom prípade z akéhokoľvek nebeského telesa) a konvertuje ho na nižší kmitočet pomocou princípu heterodýnie. Tieto skonvertovaný signál sa potom privádza po koaxiálnom vodiči na ďalšie spracovanie. Mnohokrát je skratka LNB nesprávne označovaná ako „Low Noise Blocker. Pracuje na frekvenciách cca 10 – 11,7 a 11,7 - 12.75 GHz. Oba tieto Rozsahy sú prepínané pomocou signálu 22kHz . Ovládanie polarizácie je uskutočňované jednosmerným napätím (12,5 – 14,5V pre vertikálnu polarizáciu a 15,5 – 18V pre horizontálnu).

Príloha 3 – Zdrojový kód programu pre RA os

program	writeln (a);	writeln (a);	writeln(a);	delay(b);	delay(b);
parport;	inc(j);	inc(j);	inc(j);	writeln (a);	writeln (a);
uses crt;	if j=f	if j=f	if j=f	inc(j);	inc(j);
var addr : word;	then goto	then goto	then goto	if j=f	if j=f
a,b,q,j,f,w	koniec;	koniec;	koniec;	then goto	then goto
longint;	port[addr]:= 12;	port[addr]:= 2;	if keypressed	koniec;	koniec;
x : char;	inc(a);	delay(b);	then goto	port[addr]:= 2;	port[addr]:= 12;
c : real;	delay(b);	inc(a);	koniec;	inc(a);	delay(b);
label op, uio;	writeln (a);	writeln(a);	koniec : delay	delay(b);	inc(a);
procedure	inc(j);	inc(j);	(1);	writeln (a);	writeln(a);
dopredu;	if j=f	if j=f	end;	inc(j);	inc(j);
label	then goto	then goto	procedure spat;	if j=f	if j=f
qwe, koniec;	koniec;	koniec;	label rty, koniec;	then goto	then goto
begin;	port[addr]:= 4;	port[addr]:= 3;	begin;	koniec;	koniec;
port[addr]:= 9;	inc(a);	delay (b);	port[addr]:= 1;	port[addr]:= 6;	port[addr]:= 8;
inc(a);	delay(b);	inc(a);	inc(a);	inc(a);	delay (b);
delay(b);	writeln (a);	writeln(a);	delay(b);	delay(b);	inc(a);
writeln (a);	inc(j);	inc(j);	writeln (a);	writeln (a);	writeln(a);
inc(j);	if j=f	if j=f	inc(j);	inc(j);	inc(j);
if j=f	then goto	then goto	if j=f	if j=f	if j=f
then goto	koniec;	koniec;	inc(j);	then goto	then goto
koniec;	port[addr]:= 6;	port[addr]:= 1;	if j=f	koniec;	koniec;
port[addr]:= 8;	inc(a);	delay (b);	then goto	port[addr]:= 4;	port[addr]:= 9;
inc(a);	delay(b);	inc(a);	koniec;	inc(a);	delay (b);
delay(b);			port[addr]:= 3;		
			inc(a);		

inc(a);	then goto a1;	then goto rty;	zaciatok :	then goto	then
writeln(a);	repeat;	a6: port[addr]:=	writeln ('Zadajte	koniec	port[addr]:=0
inc(j);	a1: port[addr]:=	2;	pozadovane RA	else if h='r'	else x:='g';
if j=f	9;	delay(b);	koordinaty v	then goto	if x <> 'g' then
then goto	inc(a);	inc(a);	tvare stupna s	zaciatok	goto uio;
koniec;	delay(b);	writeln(a);	desatinnymi	end;	if x = 'g'
if keypressed	writeln (a);	if keypressed	miestami);	koniec ; ;	then goto op;
then goto	if keypressed	then goto rty;	readln (d);	end ;	writeln
koniec;	then goto rty;	a7: port[addr]:=	if (d<0) OR		('hotovo');
koniec: end;	a2: port[addr]:=	3;	(d>360)		readln;
procedure	8;	delay (b);	then begin		port[addr]:=0;
celestial;	inc(a);	inc(a);	writeln('hodnota		end. ;
label a1, a2, a3,	delay(b);	writeln(a);	mimo rozsah');	begin;	
a4, a5, a6, a7,	writeln (a);	if keypressed	goto zaciatok;	c:=0;	
a8;	if keypressed	then goto rty;	end;	addr:=MemW[\$	
label rty;	then goto rty;	a8: port[addr]:=	e:=abs(c-d);	0040:\$0008];	
begin;	a3: port[addr]:=	1;	f:= round	writeln (addr);	
q:=port[addr];	12;	delay (b);	(e/0.077922077	port[addr]:=0;	
w:=b ;	inc(a);	inc(a););	op: readln (b);	
b:=18650 ;	delay(b);	writeln(a);	g:=0;	uio : x :=	
if q = 9	writeln (a);	if keypressed	if c>d	readkey;	
then goto a2	if keypressed	then goto rty;	then	if x = 's'	
else if q= 8	then goto rty;	until 5=6;	begin repeat	then spat	
then goto a3	a4: port[addr]:=	rty : delay (1);	spat ;	else if x= 'd'	
else if q=12	4;	b:=w;	until (j=f) or	then dopredu	
then goto a4	inc(a);	end;	(j>f);	else if x='h'	
else if q=4	delay(b);	procedure	end;	then moveto	
then goto a5	writeln (a);	moveto;	c:=d;	else if x= 'a'	
else if q=6	if keypressed	var h : char;	celestial;	then	
then goto a6	then goto rty;	var g : longint;	if keypressed	begin	
else if q=2	a5: port[addr]:=	var e,d : real;	then begin	port[addr]:=0;	
then goto a7	6;	inc(a);	halt;	end	
else if q=3	inc(a);	delay(b);	if h = 'e'	else if x='f'	
then goto a8	writeln (a);	writeln (a);			
else if q=1	if keypressed	label			
		zaciatok, koniec			
		c;			
		begin;			

Príloha 4 – Zdrojový program pre ovládanie DEC osi

```

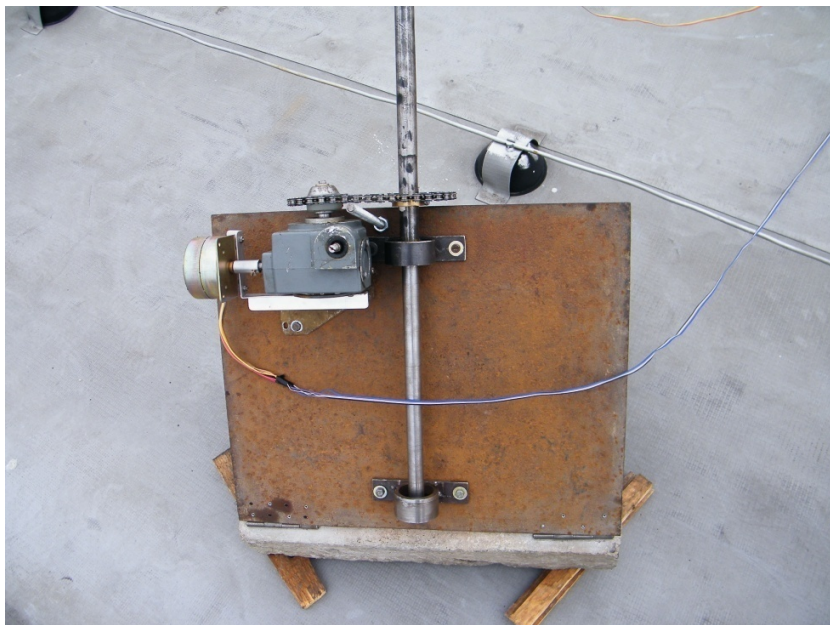
program servo;          f:=round(e/0.06      delay(2);          then          until          port[addr]:=0;
uses crt;              0595859) ;          inc(g);          begin;          keypressed;          clrscr;
var addr : word;      g:=0;              until f=g;          repeat          end          writeln('0');
a :char;              if c>d            port[addr]:=0      port[addr]:=96;    else if a =b'      end
b,f,g : longint;     then            end;              delay(2);          then begin;          else if a='m'
c,d,e : real;        begin            c:=d;              until          repeat          then
label starter;       repeat          goto zaciatok;    keypressed;        port[addr]:=160;   begin;
procedure           port[addr]:=160;    end;              end                delay(2);          port[addr]:=0;
celestial;          delay(2);          begin;            else if a='t'      dec(b);
label zaciatok;     inc(g);            b:=0;              then celestial    writeln(b);
begin;              until f=g;        c:=0;              else if a='v'      until
zaciatok :          port[addr]:=0      clrscr;            then begin;        keypressed;        end;
writeln ('zadajte   end          addr:=MemW[$       repeat          end          until 5=6;
pozadovane         else if d>c      0040:$0008];     port[addr]:=128;  else if a= 'n'      end.;
DEC koordinaty     then            starter : repeat; delay(2);          then          port[addr]:=0
v tvare uhla v     begin;          begin;            inc(b);          else if a=' '
desiatkovaj       repeat          a:=readkey;      writeln(b);        then begin
sustave');        port[addr]:=128;  if a='c'          b:=0          ;
readln (d);        begin          begin;            inc(b);          else if a=' '
d:=(d+90);         repeat          a:=readkey;      writeln(b);        then begin
e:=abs(c-d);       port[addr]:=128;  if a='c'          b:=0          ;

```

Príloha 5 – Riešenie montáže



Na obrázku vidieť parabol, v popredí s LNB, vpravo dole je upevnené na deklinačnej osi protizávažie. Prevod z pohonnej jednotky na os je zabezpečený bicyklovou reťazou. Na obrázku sa nachádza aj koaxiálny vodič a plochý farebný vodič určený na napájanie pohonov.

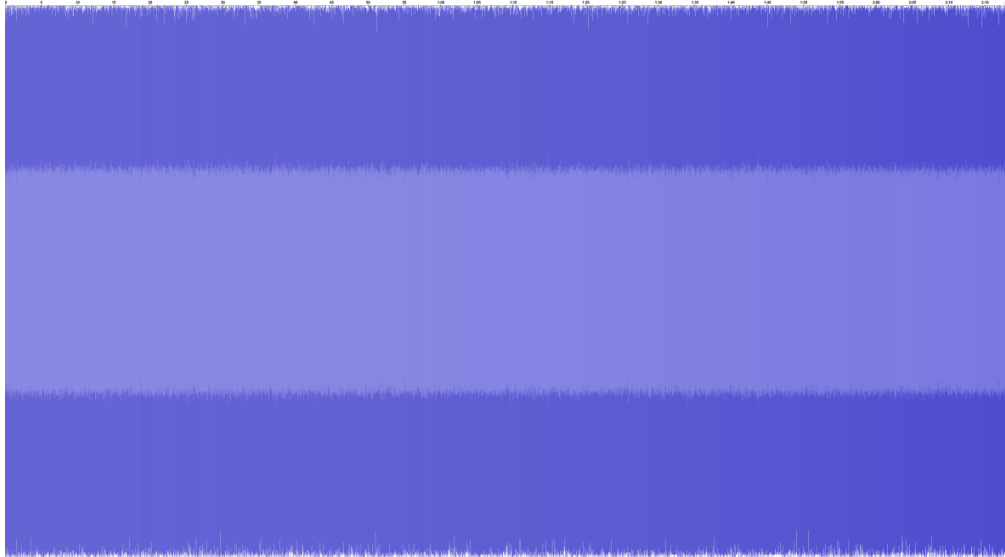


Upevnenie deklinačnej osi pomocou guľôčkových ložísk. Aj v tomto prípade je prenos hnacej sily z pohonnej jednotky na os zabezpečený bicyklovou reťazou.



Autorský tím zo svojím výtvorom.

Príloha 6 – Grafy

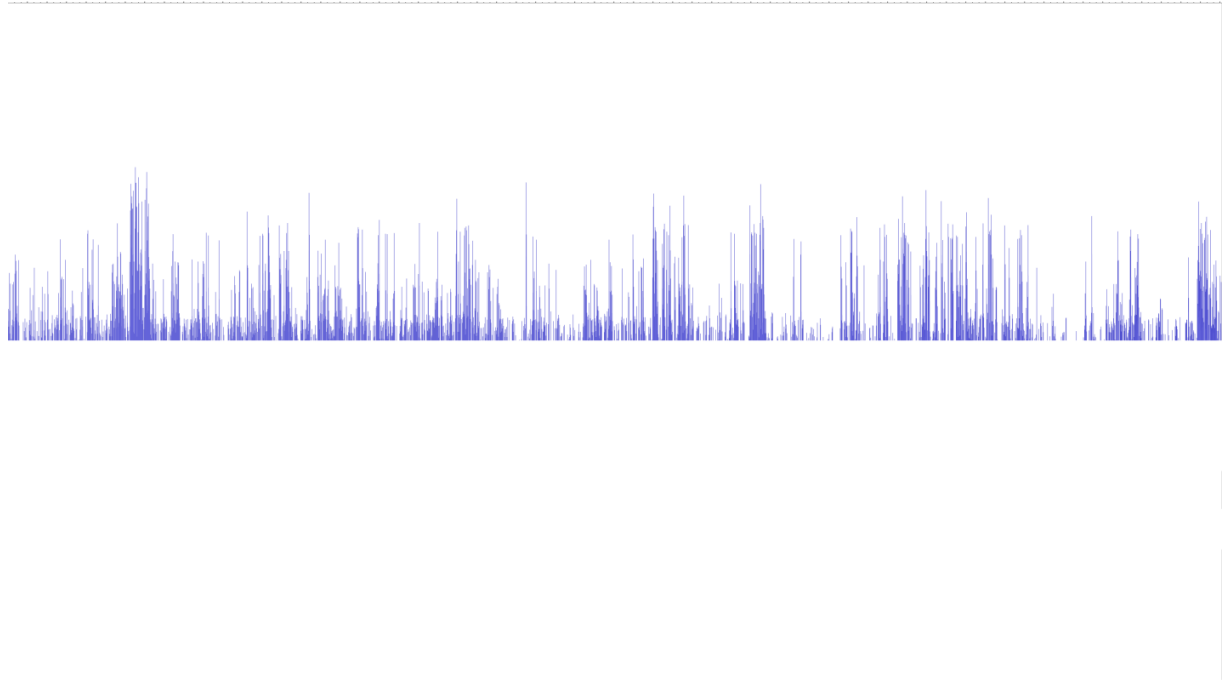


Graf zobrazujúci snímku Slnka ktorá ešte nemá odstránený šum.



Snímka referenčného šumu.

0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 1450 1500 1550 1600 1650 1700 1750 1800 1850 1900 1950 2000 2050 2100 2150 2200 2250 2300 2350 2400 2450 2500 2550 2600 2650 2700 2750 2800 2850 2900 2950 3000 3050 3100



Výsledok po odstránení šumu. Graf predstavuje slnečnú aktivitu v priebehu 30 minút.