

```

'{$STAMP BS2}
'{$PBASIC 2.5}

'motor number 0,1 - BASE
'motor number 2,3 - Top

adcbits      VAR      Byte
angle        VAR      Byte
angle1       VAR      Word
choose       VAR      Bit

initial:
'auxio

HIGH 0
LOW 0
LOW 1
PULSOUT 1, 210
SHIFTIN 2, 1, MSBPOST, [adcbits\8]
IF adcbits <= 3 THEN GOTO moving_down
IF adcbits <= 256 THEN moving_down
PAUSE 100

'initial1:
'HIGH 10
'LOW 10
'LOW 11
'PULSOUT 11, 210
'SHIFTIN 12, 11, MSBPOST, [adcbits\8]
'IF adcbits <= 3 THEN prepare
'IF adcbits <= 256 THEN moving_down

moving_down:
GOSUB check_top_ADC
IF adcbits <= 7 THEN moving_counterclock

SEROUT 4,84,[$80,0,3,55]
PAUSE 20

IF adcbits <= 7 THEN GOSUB stop_moving
GOTO moving_down

moving_counterclock:
GOSUB check_bot_ADC
'DEBUG ? adcbits
IF adcbits <= 5 THEN prepare

SEROUT 4,84,[$80,0,1,55]
PAUSE 20

IF adcbits <= 5 THEN GOSUB stop_moving
GOTO moving_counterclock

prepare:
GOSUB stop_moving

```

```

DEBUG CLS
DEBUG "insert tilt angle betw. 0 & 45" ,CR
DEBUGIN DEC angle
IF (angle > 0) & (angle < 45) THEN other
GOTO prepare

other:
DEBUG CR, CR,"insert rotating angle betw. 0 & 360" ,CR
DEBUGIN DEC angle1
IF (angle1 > 0) & (angle1 <360) THEN main
GOTO other

main:
angle = (angle * 51)/ 9          ' bottom
DEBUG "ADC_value for Top " , ? angle
IF angle = 0 THEN prepare

angle1 = (angle1 * 51) /72
DEBUG CR, "ADC_value for Bottom " , ? angle1
IF angle1 = 0 THEN prepare

moving_upward:
HIGH 4
LOW 5
HIGH 5
PAUSE 10
GOSUB check_top_ADC

IF ((angle) - 4) <= adcbits AND ((angle)+ 4) >= adcbits THEN
moving_clockwise

SEROUT 4,84,[$80,0,2,66]
PAUSE 20

IF ((angle) - 4) <= adcbits AND ((angle) + 4) >= adcbits THEN GOSUB
stop_moving
IF adcbits = 255 THEN GOSUB stop_moving
IF adcbits =255 THEN prepare
GOTO moving_upward

moving_clockwise:
HIGH 4
LOW 5
HIGH 5
PAUSE 10
GOSUB check_bot_ADC

IF (angle1 -2) <= adcbits AND (angle1 +2) >= adcbits THEN Finish

SEROUT 4,84,[$80,0,0,66]          'shaft rotates clockwise
PAUSE 20

IF (angle1 -2) <= adcbits AND (angle1 +2 ) >= adcbits THEN GOSUB
stop_moving
GOTO moving_clockwise

```

```
Finish:
DEBUG CLS, "End"
DEBUG CR, "restart press 1"
DEBUGIN DEC choose
IF choose = 1 THEN initial
GOTO Finish
END
```

```
check_top_ADC:
DEBUG CLS
HIGH 0
LOW 0
LOW 1
PULSOUT 1, 210
SHIFTIN 2, 1, MSBPOST, [adcbits\8]
DEBUG CR, "Value for Top ADC  ", ? adcbits
RETURN
```

```
check_bot_ADC:
DEBUG CLS
HIGH 10
LOW 10
LOW 11
PULSOUT 11, 210
SHIFTIN 12, 11, MSBPOST, [adcbits\8]
DEBUG CR, "Value FOR Bottom ADC  ", ? adcbits
RETURN
```

```
stop_moving:
SEROUT 4,84,[$80,0,1,0]
SEROUT 4,84,[$80,0,3,0]
PAUSE 20
RETURN
```