

AN002 - Interfacing Inclination Sensors to the Chickadee XL SBC

Topic

This application note describes how to interface an inclination sensor to the Chickadee XL SBC. There are various techniques for measuring inclination, but the emphasis here is on medium accuracy, low complexity, and low cost. This inclination application covers a range of $\pm 20^\circ$, but is easily adaptable to other ranges.

The Challenge

The inspiration for this application note was to protect a trailer-mounted machine from running without proper oil lubrication. The oil system of the machine consisted of an oil pan with a sump type pump. Due to the oil system capacity and mechanical design the acceptable range of inclination for this machine was $\pm 10^\circ$ in one axis and $\pm 20^\circ$ in the perpendicular axis. Any inclinations beyond these could result in insufficient pooling of oil in the oil pan, resulting in oil delivery failure.

The Solution

Modern sensor technology has greatly simplified the sensing of inclination. The Lucas Sensing Systems Accustar II Dual Axis Clinometer offers an elegant, simple, low cost, and complete solution to inclination sensing. In this application note the following items are used:

- 1 Chickadee XL SBC with ADC option (Bagotronix)
- 1 Accustar II Dual Axis Clinometer part# 02753-01 (Lucas Sensing Systems)
- 1 Matching connector for Accustar II part# 09-01-1061A (Lucas Sensing Systems)

The schematic is shown in Figure 1.

Theory of Operation

The clinometer analog outputs for X and Y axes are connected to the Chickadee ADC channels 0 and 2, respectively. The clinometer PWM outputs are not used. The clinometer analog outputs are ratiometric with its supply voltage, so either a well-regulated power supply should be used, or the Chickadee ADC should be set up for ratiometric operation and powered from the same supply the clinometer uses. In this application note, we used the Chickadee ADC in absolute operation (0 to 4.095 V). We also used the Chickadee voltage regulator option to supply the Chickadee and the clinometer with a well-regulated supply voltage. The Chickadee regulator is adjustable with the on-board trimpot to exactly +5V.

Since the clinometer runs on +5V, it seems that a possibility exists to overdrive the ADC inputs. In practice, the actual characteristic of the clinometer is that the maximum output voltage is about +4V, due to the "headroom" requirements of the internal electronics. Therefore, no protection circuitry is needed on the ADC inputs.

When operated from a power supply of +5V, the clinometer has a zero-angle output of 2.5V on each axis. In terms of unsigned binary ADC weight, this equates to 2500 (decimal) and 9C4 (hexadecimal). This offset must be subtracted from each axis to center each axis around zero. Axis readings greater than this offset will give positive angles; less than this offset will give negative angles. After the offset is removed, the axis must be scaled by dividing the signed binary value by the scale factor, which is in millivolts per degree. For operation from +5V, this is about $56\text{mV}/^\circ$.

AN002 - Interfacing Inclinometers to the Chickadee XL SBC

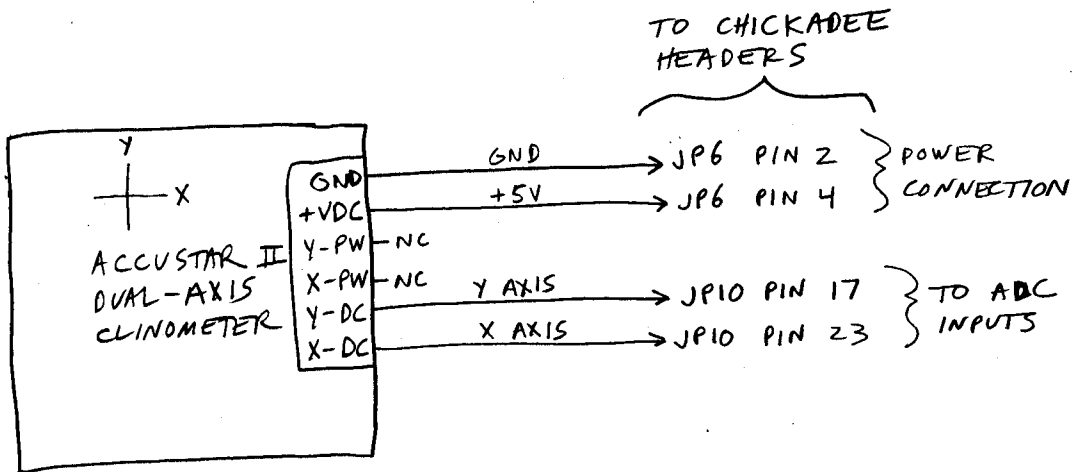


Figure 1

The Software

The clinometer interface software was written in Borland C v3.0. Other C compilers and versions can be used, although some minor modifications may be required. The program consists of the following files, compiled and linked to produce a 16-bit DOS EXE:

- CSFCADC.C
- CHICKADE.C
- CHKTILT.C

Using the TRANSFER.EXE utility, the program CHKTILT.EXE was transferred to the Chickadee XL flash disk. Then the program was run and the clinometer was tilted over various angles:

```
B>CHKTILT
```

Strike the 'X' key to quit, any other alphanumeric key to sample.

```
X axis: 0  
Y axis: 0
```

```
X axis: -13  
Y axis: 0
```

```
X axis: -19  
Y axis: 0
```

```
X axis: -20  
Y axis: -1
```

```
X axis: 6  
Y axis: 1
```

```
X axis: 14  
Y axis: 1
```

AN002 - Interfacing Inclination Sensors to the Chickadee XL SBC

X axis: 19
Y axis: 0

X axis: 23
Y axis: 0

X axis: 23
Y axis: 0

X axis: 22
Y axis: 1

X axis: 21
Y axis: 1

X axis: 20
Y axis: 1

X axis: 1
Y axis: 8

X axis: 0
Y axis: 10

X axis: 0
Y axis: 13

X axis: 0
Y axis: 15

X axis: 0
Y axis: 17

X axis: 0
Y axis: 18

X axis: 0
Y axis: 19

X axis: 0
Y axis: 20

X axis: 0
Y axis: -1

X axis: 0
Y axis: -5

X axis: 0
Y axis: -10

X axis: 0
Y axis: -19

X axis: 0
Y axis: -4

X axis: 0
Y axis: 0

X axis: 0

AN002 - Interfacing Inclination Sensors to the Chickadee XL SBC

Y axis: 0

B>

Sometimes the inclination reading is beyond the clinometer's range. This corresponds to extreme inclinations beyond +/-20°.

The source file for the CHKTILT.EXE program is listed here and is also available with the other source files in ZIPped form from the Bagotronix website.

Program Listing

```
/******  
Bagotronix  
1019 Crossing Brook Way  
Tallahassee, FL 32311  
  
(C) Copyright 1998 Bagotronix  
All rights reserved  
  
Filename:      chktilt.c  
Programmer(s): Ivan Baggett  
Description:   Chickadee(tm) example application for inclination sensing.  
comments:     Uses Lucas Sensing Systems Accustar II  
              dual axis clinometer (tilt sensor).  
              Chickadee ADC is set up for absolute 0 - 4.095 V range.  
*****/  
  
#include <dos.h>  
#include <stdio.h>  
  
#include "stdinc.h"  
#include "chickade.h"  
#include "csfcadc.h"  
  
#define _getch getch          // for Borland console  
  
// scale factor for operating Accustar II on +5V supply (mV per degree)  
#define TILT_SCALE 56  
#define TILT_OFFSET 0x9c4  
  
int main (void) {  
  
    BOOLEAN done = FALSE;  
    int c, i;  
    UWORD u;  
    UBYTE b;  
  
    printf ("Strike the 'X' key to quit, any other alphanumeric key to sample.\n\n");  
  
    while (!done) {  
        while (!kbhit());  
        c = _getch();  
        if (c == 'x' || c == 'X')  
            done = TRUE;  
  
        printf ("\n");  
        // X axis is on ADC channel 0  
        // Y axis is on ADC channel 2  
        for (i = 0; i <= 2; i += 2) {  
            u = ADCget (MAX186, (UBYTE)i, 0);    /* MAX186, channel i, 0 acq delay */  
            /* 12-bit ADC result is normalized to MSB, so denormalize it (/16) */  
            u /= 16;  
  
            c = (((int)u * 2 - TILT_OFFSET * 2) + 1) / 2) / TILT_SCALE;  
            printf ("%c axis: %d\n", 'X' + i / 2, c);    /* to remote console */  
        }  
    }  
  
    return 0;  
}
```