Using Eagle Tree Microsensor V3/V4 Sensors with your Microcontroller Project

Document Version 1.5

The Eagle Tree Systems G-Force, Airspeed, Altimeter V4, and Altimeter V3 sensors can be incorporated in you microcontroller project. The sensors are accessed via the industry standard I2C bus.

Data Format Returned

Altimeter V3 and Airspeed V3

When put in Third Party mode described below, the sensors return the actual speed or altitude, in the units selected as described in the instruction manual that accompanies the sensors. Note that the Altimeter V3 never reports below 0.

Altimeter V4

The Altimeter V4 reports altitude in decimeters (tenths of meters). The value returned is signed, and is the actual barometric altitude, + 3000 decimeters. Subtract 3000 from the returned value to determine the actual barometric altitude, or just use the initial returned value as the zero reference point. The sensor will report negative altitude readings.

G-Force

The G-Force Microsensors return signed centi-*g* (hundredths of *g*) readings for X, Y and Z (in that order).

Configuring the Airspeed V3 and Altimeter V3 Sensors for “Third Party” Mode

The Airspeed V3 and Altimeter V3 Microsensors need to be configured for “Third Party” mode to use with your project. The Altimeter V4 and G-Force sensors do not have this requirement.

To configure the sensor to be accessible with your Microcontroller, you must first connect the sensor to an eLogger V3/V4, and install Eagle Tree Data Recorder software version 9.57 or later. Follow these steps to switch the sensor from “Eagle Tree” mode (raw data returned via I2C) to “Third Party” mode (actual airspeed or altitude returned via I2C):

1. Install Data Recorder software version 9.57 or later.
2. Attach the MicroSensor to your eLogger V3 or V4, per the Microsensor instructions.
3. Connect the eLogger V3/V4 and Microsensor to USB.
4. In the Data Recorder software click "Advanced, Custom Hardware Options."
5. Click the appropriate checkbox (“Use my Airspeed V3 Microsensor with 3rd Party hardware” or “Use my Altimeter V3 Microsensor with 3rd Party hardware “ to toggle the sensor(s) between Eagle Tree and third party mode.

6. When the checkbox is checked, the sensor is placed in Third Party mode. Likewise, when the checkbox is cleared, the sensor is reverted back to Eagle Tree mode. **Note that the data logged with Eagle Tree equipment will ONLY be correct if the sensor is placed in Eagle Tree mode**.
7. When the sensor is in Third Party mode, the sensor returns the computed value, in the units programmed into the sensor by the user. In other words, if the airspeed sensor is programmed to read MPH, the value returned is in MPH. See the sensor manual on configuring the units (units cannot be configured via I2C).

Connecting the MicroSensor to your Microcontroller

The pinout of the MicroSensor’s 4 pin cable is as follows:

- Red wire: Power (3V to 16V) (3V to 7V for High G G-Force Microsensor)
- White wire: Ground
- Yellow wire: SDA (3V to 5V)
- Brown wire: SCL (3V to 5V)

Note that pull-ups are required on the SDA and SCL lines, and are not provided on the MicroSensor. These must be added t o your PC board.

Accessing the Microsensors via I2C

For the airspeed sensor, the I2C address (SENSOR_ADDRESS in the pseudo code below) is **0xEA**
For the altimeter V3 sensor, the I2C address (SENSOR_ADDRESS in the pseudo code below) is **0xE8**
For the altimeter V4 sensor, the I2C address (SENSOR_ADDRESS in the pseudo code below) is **0xEC**
For the GForce sensors, the I2C address (SENSOR_ADDRESS in the pseudo code below) is **0xA8**

I2C access should work correctly between 100Khz up to 400Khz. It may be possible to access the sensor at slower or higher speeds, however.

Here is the pseudo code for accessing the Airspeed and Altimeter sensors:

```
byte data[2];
signed short reading = 0xFFFF;

i2c_start();

// select sensor in write mode

if (!(i2c_write(SENSOR_ADDRESS | I2C_WRITE_BIT))) {

        // send "read data" command to sensor

        if (!i2c_write(0x07)) {
```

```
        i2c_restart();  // perform I2C restart

        // select sensor in read mode

        if (!i2c_write(| SENSOR_ADDRESS | I2C_READ_BIT)) {

            // read two bytes of sensor data

            data[0] = i2c_read(1);
            data[1] = i2c_read(0);

            reading = *((signed short *)(&data[0]));
        }
    }
}

i2c_stop();
```

Here is the pseudo code for accessing the G-Force sensors:

```
byte data[6];
signed short GX, GY, GZ;
i2c_start();

// select sensor in write mode

if (!(i2c_write(SENSOR_ADDRESS | I2C_WRITE_BIT))) {

    // send "read data" command to sensor

    if (!i2c_write(0x07)) {

        i2c_restart();  // perform I2C restart

        // select sensor in read mode

        if (!i2c_write(| SENSOR_ADDRESS | I2C_READ_BIT)) {

            // read 6 bytes of sensor data

            data[0] = i2c_read(1);
            data[1] = i2c_read(1);
            data[2] = i2c_read(1);
            data[3] = i2c_read(1);
            data[4] = i2c_read(1);

            data[5] = i2c_read(0);
```

```
                                GX = *((signed short  *)(&data[0]));
                                GY = *((signed short  *)(&data[2]));
                                GZ = *((signed short *)(&data[4]));

                        }
                }
        }

        i2c_stop();
```

Troubleshooting I2C Access to the Microsensor

If you have an issue accessing the sensor with your microcontroller, post a support question on our RCGroups support forum at this location:

http://www.rcgroups.com/forums/showthread.php?t=1118096

Note that email, telephone or web ticket support is not provided for accessing the sensors via I2C.