

Due to the Observation on page 92, we are trying the BNO 085 and using a different library until we can determine how to fix the BNO 085.

"BNO.h" Header File

```
1  #ifndef BNO_H
2  #define BNO_H
3
4  #include <Adafruit_BNO08x_RVC.h>
5
6  class BNO
7  {
8      private:
9          int bnoRST;
10         Stream *streamPtr;
11         Adafruit_BNO08x_RVC myIMU;
12         BNO08x_RVC_Data bnoData;
13     public:
14         BNO();
15         bool setupBNO(Stream *theSerial, int aResetPin);
16         double GetHeading();
17     };
18
19 #endif
```

BNO.cpp:

```
1  #include "BNO.h"
2
3  BNO::BNO(){}
4
5  bool BNO::setupBNO(Stream *theStream, int aResetPin)
6  {
7      myIMU = Adafruit_BNO08x_RVC();
8      streamPtr = theStream;
9      bnoRST = aResetPin;
10
11     if (!myIMU.begin(theStream))
12         return false;
13
14     pinMode(bnoRST, OUTPUT);
15     digitalWrite(bnoRST, LOW);
16     delay(10);
17     digitalWrite(bnoRST, HIGH);
18     return true;
19 }
```

Here, we implement an empty constructor and instead let the user call the "setupBNO" method.

Notice that we are really just "wrapping" Adafruit's library with our own extra functionality.

```
20
21 double BNO::GetHeading()
22 {
23
24     while (streamPtr->available() > 0)
25     {
26         streamPtr->read();
27     }
28
29     // Continue reading until we get a new heading.
30     while(!myIMU.read(&bnoData)) {}
31
32     // Adjust magnetic reading to get geographic north.
33     // float heading = bnoData.yaw - declination;
34
35     // Adjust using GPS offset
36     double heading = bnoData.yaw; //+ headingOffset;
37
38     // Make sure we get a value between 0-360
39     while (heading < 0)
40     {
41         heading += 360;
42     }
43
44     while (heading > 360)
45     {
46         heading -= 360;
47     }
48
49     return heading;
50 }
```

Here, we implement how to get the heading.

Sample Driver Code and Out put:

```

1 #include "src/BNO/BNO.h"
2
3 BNO myBNO = BNO();
4 int bnoResetPin = 6;
5 double currentHeading;
6
7 void setup()
8 {
9     // Local Debugging Serial monitor
10    Serial.begin(9600);
11    while(!Serial);
12    Serial.println("#GetObsessed!");
13
14    // Start the BNO on Serial2 with the reset pin
15    Serial2.begin(115200);
16    if (!myBNO.setupBNO(&Serial2, bnoResetPin))
17    {
18        digitalWrite(LED_BUILTIN, HIGH);
19        Serial.println("BNO Failed to start");
20        while(1);
21    }
22 }
23
24 void loop()
25 {
26     currentHeading = myBNO.GetHeading();
27     Serial.println(currentHeading);
28     delay(500);
29 }

```

```

13:27:27.888 -> #GetObsessed!
13:27:27.888 -> 359.98
13:27:28.408 -> 359.98
13:27:28.921 -> 359.98
13:27:29.442 -> 359.98
13:27:29.915 -> 359.97
13:27:30.437 -> 359.97
13:27:30.952 -> 353.39
13:27:31.472 -> 348.17
13:27:31.967 -> 341.65
13:27:32.479 -> 350.21
13:27:32.976 -> 7.96
13:27:33.498 -> 9.86
13:27:34.020 -> 9.60
13:27:34.542 -> 9.23
13:27:35.050 -> 9.23

```