

# IVN\_lib

Arduino compatible library for the VN300 IMU. Designed with the Teensy 3.2 & 4.0 specifically in mind.

User modifiable #defines are found in IVN\_user.h. It is particularly important to uncomment the #define that pertains to the Serial port that you are using. Serial1 is default.

All page references are to the VN300 User Manual found: [https://www.vectornav.com/docs/default-source/documentation/vn-300-documentation/vn-300-user-manual-\(um005\).pdf](https://www.vectornav.com/docs/default-source/documentation/vn-300-documentation/vn-300-user-manual-(um005).pdf)

## Basic 4 startup steps for use:

1. Initialize IVN\_IMU object before void setup(). "VN300" will be used throughout this document.

```
IVN_IMU VN300; // Initialization of VN300 IMU object.
```

2. Need two variables for holding message data

```
Binary_OutGroup_Type VNoutGroup; // Make variable for holding Group and Field settings for outgoing messages (to VN300)
```

```
VN_Message_Type VNoutMessage; // Make variable for holding VN Serial Message info for incoming messages (from VN300)
```

3. Begin serial communication within setup().

```
VSerial.begin(VN100_Baud_115200); // Begin Serial with IMU
```

4. Start IMU with any of the following antenna offsets within setup().

### A. Simplest with no offset:

```
VN300.begin(); // Begin with just Async Output enabled
```

### B. Only offset for antenna 1 (GNSS1) given in meters

```
VN300.begin(1.0,0.0,1.0); // example as 1 meter, 0 meters, & 1 meter
```

### C. Offsets for both antennae (GNSS1 & 2) given in meters with default uncertainty of 0.0254

```
VN300.begin(1.0,0.0,1.0,-2.0,0.0,1.2);
```

### D. Offsets for both antennae (GNSS1 & 2) given in meters with user specific uncertainty

```
VN300.begin(1.0,0.0,1.0,-2.0,0.0,1.2, 0.035, 0.035, 0.35);
```

### E. Offsets for both antennae (GNSS1 & 2) given in meters with user specific uncertainty in user defined array variable. "beginGNSSdata" defined before setup() as "float beginGNSSdata[9] = {1.0, 0.0, 1.0, -2.0, 0.0, 1.2, 0.035, 0.035, 0.35};"

```
VN300.begin(beginGNSSdata);
```

## Optional 5. Setup Serial communication to use 8-bit CRC and error reporting (within setup()) p98

```
int comProControlArray[7] = {0, 0, 0, 0, 1, 0, 1};
```

```
VN300.writeSerialReg(VN_REG_ComProControl, comProControlArray);
```

## IMU data Output:

There are several ways to have the VN300 provide IMU data. Up to three sets (Binary Outputs 1, 2 & 3) of the following Binary Outputs can be setup to either send the data automatically at certain rates, or stored so they can be polled whenever desired. Those three sets can be any combination of Regular or Polled Outputs. When read by your program, the IMU data will be found in:

"VN300.VN\_Message\_Binary.Binary\_Message\_Payload" "VN300" needs to be changed to whatever your initialized object was named. List of available Serial options and Hz rate options can be found in IVN\_lib.h, line 119.

### 1. Regular Binary Output. Example Sketch is SimpleSerialRead.ino.

#### A. Clear variable for new Binary Output command

```
VNoutGroup = VN300.clearGroup(VNoutGroup);
```

#### B. Set the desired fields to true (at least 1, but many can be set)

```
VNoutGroup.OutGroupIMU.OutField6 = true; // Set Group3, Field6 (DeltaTheta)
```

```
VNoutGroup.OutGroupIMU.OutField7 = true; // Set Group 3, Field7 (DeltaVel)
```

#### C. Send actual command

```
VN300.startBinOutput(1, SerialOne, Rate20Hz, VNoutGroup); // Start Async Binary Output 1 on  
VN-300 Serial 1, updating at 20Hz. The IMU data will be sent via the selected VN300's Serial  
Port at the selected rate
```

### 2. Polled Binary Output. Example Sketch is SimplePollRead.ino.

#### A. Clear variable for new Binary Output command

```
VNoutGroup = VN300.clearGroup(VNoutGroup); // Clear variable for new Polling command
```

#### B. Set the desired fields to true (at least 1, but many can be set)

```
VNoutGroup.OutGroupCommon.OutField3 = true; // Set Group1, Field3 (YawPitchRoll)
```

```
VNoutGroup.OutGroupIMU.OutField4 = true; // Set Group 3, Field4 (Temp)
```

#### C. Send actual command

```
VN300.startPoll(1, Rate10Hz, VNoutGroup); // Start Polled Binary Output 1 updating at 10Hz  
(rate doesn't really matter). The IMU data will be updated regularly and available to be read via  
the register.
```

The VN300 can also send IMU data whenever requested by reading a register anytime via the command

```
VN300.readSerialReg(NameofRegister);
```

When read by your program, the IMU data will be found in:

"VN300.VN\_Message\_ASCII.ASCII\_Message\_Payload" "VN300" needs to be changed to whatever your initialized object was named. List of available registers can be found in IVN\_lib.h, line 10.

IVN\_lib.h also notes the page in the User Manual where the listed register can be found.

### **All available commands:**

```
public: void readSerialReg(int register2Read);
public: void writeAsyncPause(int setting); //Pause or Resume all Serial output from VN300
public: void readPoll(int channel); //Read IMU data setup via Polled Binary output
public: void writeSerialReg(int register2Write, int data[]); //Write to IMU Register
public: void writeSerialReg(int register2Write, float data[]); //Write to IMU Register
public: void writeFactoryReset(); //Reset IMU to factory settings
public: void writeReset(); //Reset IMU
public: void writeSave(); //Write and Save settings for next power on
public: void writeFirmUpdate(); //Start IMU firmware update
public: void writeSerialPrompt(); //Start human direct serial command prompt
public: void writeUserTag(char userTag[]); //Add UserTag to IMU
public: void checkVNSerial(char inChar, VN_Message_Type &VN_Message); //Collects Serial data from IMU for
program use
```

### **CRC data:**

Although CRC data is collected. It is not automatically checked by the library. All library commands are currently sent with the CRC set to "XX".

### **VN\_Message\_Type structure explained:**

VN\_Message\_Type needs to be declared before setup(). Examples will use:

```
VN_Message_Type VNoutMessage; // Make variable for holding VN Serial Message info
```

```
VNoutMessage // user defined variable name
```

```
    .VN_Message_ASCII // Set of ASCII received data
```

```
        .ASCII_Message_Payload // String holding received data
```

```
        .ASCII_Message_CRC // unsigned int holding CRC
```

```
    .VN_Message_Binary // Set of Binary Received data
```

```
        .Binary_Message_Group // unsigned char which OutGroup has data
```

```
        .Binary_Message_Fields // Set of Received Field data
```

```
            .OutGroupCommon // Set of Received Field data for Group1, "Common"
```

```
                .OutField0 // boolean true if there is data for this field
```

```
                .OutField1 // boolean true if there is data for this field
```

```
                .OutField2 // boolean true if there is data for this field
```

```
                .OutField3 // boolean true if there is data for this field
```

```
.OutField4 // boolean true if there is data for this field
.OutField5 // boolean true if there is data for this field
.OutField6 // boolean true if there is data for this field
.OutField7 // boolean true if there is data for this field
.OutField8 // boolean true if there is data for this field
.OutField9 // boolean true if there is data for this field
.OutField10 // boolean true if there is data for this field
.OutField11 // boolean true if there is data for this field
.OutField12 // boolean true if there is data for this field
.OutField13 // boolean true if there is data for this field
.OutField14 // boolean true if there is data for this field
.OutField15 // boolean true if there is data for this field
.OutFields // unsigned int that can access all the Field bits at once
.OutGroupTime // Set of Received Field data for Group2, "Time"
.OutField0 // boolean true if there is data for this field
.OutField1 // boolean true if there is data for this field
.OutField2 // boolean true if there is data for this field
.OutField3 // boolean true if there is data for this field
.OutField4 // boolean true if there is data for this field
.OutField5 // boolean true if there is data for this field
.OutField6 // boolean true if there is data for this field
.OutField7 // boolean true if there is data for this field
.OutField8 // boolean true if there is data for this field
.OutField9 // boolean true if there is data for this field
.OutField10 // boolean true if there is data for this field
.OutField11 // boolean true if there is data for this field
.OutField12 // boolean true if there is data for this field
.OutField13 // boolean true if there is data for this field
.OutField14 // boolean true if there is data for this field
.OutField15 // boolean true if there is data for this field
.OutFields // unsigned int that can access all the Field bits at once
.OutGroupIMU // Set of Received Field data for Group3, "IMU"
.OutField0 // boolean true if there is data for this field
.OutField1 // boolean true if there is data for this field
```

```
.OutField2 // boolean true if there is data for this field
.OutField3 // boolean true if there is data for this field
.OutField4 // boolean true if there is data for this field
.OutField5 // boolean true if there is data for this field
.OutField6 // boolean true if there is data for this field
.OutField7 // boolean true if there is data for this field
.OutField8 // boolean true if there is data for this field
.OutField9 // boolean true if there is data for this field
.OutField10 // boolean true if there is data for this field
.OutField11 // boolean true if there is data for this field
.OutField12 // boolean true if there is data for this field
.OutField13 // boolean true if there is data for this field
.OutField14 // boolean true if there is data for this field
.OutField15 // boolean true if there is data for this field
.OutFields // unsigned int that can access all the Field bits at once
.OutGroupGNSS1 // Set of Received Field data for Group4, "GNSS1"
    .OutField0 // boolean true if there is data for this field
    .OutField1 // boolean true if there is data for this field
    .OutField2 // boolean true if there is data for this field
    .OutField3 // boolean true if there is data for this field
    .OutField4 // boolean true if there is data for this field
    .OutField5 // boolean true if there is data for this field
    .OutField6 // boolean true if there is data for this field
    .OutField7 // boolean true if there is data for this field
    .OutField8 // boolean true if there is data for this field
    .OutField9 // boolean true if there is data for this field
    .OutField10 // boolean true if there is data for this field
    .OutField11 // boolean true if there is data for this field
    .OutField12 // boolean true if there is data for this field
    .OutField13 // boolean true if there is data for this field
    .OutField14 // boolean true if there is data for this field
    .OutField15 // boolean true if there is data for this field
    .OutFields // unsigned int that can access all the Field bits at once
.OutGroupAttitude // Set of Received Field data for Group5, "Attitude"
```

```
.OutField0 // boolean true if there is data for this field
.OutField1 // boolean true if there is data for this field
.OutField2 // boolean true if there is data for this field
.OutField3 // boolean true if there is data for this field
.OutField4 // boolean true if there is data for this field
.OutField5 // boolean true if there is data for this field
.OutField6 // boolean true if there is data for this field
.OutField7 // boolean true if there is data for this field
.OutField8 // boolean true if there is data for this field
.OutField9 // boolean true if there is data for this field
.OutField10 // boolean true if there is data for this field
.OutField11 // boolean true if there is data for this field
.OutField12 // boolean true if there is data for this field
.OutField13 // boolean true if there is data for this field
.OutField14 // boolean true if there is data for this field
.OutField15 // boolean true if there is data for this field
.OutFields // unsigned int that can access all the Field bits at once
.OutGroupINS // Set of Received Field data for Group6, "INS"
.OutField0 // boolean true if there is data for this field
.OutField1 // boolean true if there is data for this field
.OutField2 // boolean true if there is data for this field
.OutField3 // boolean true if there is data for this field
.OutField4 // boolean true if there is data for this field
.OutField5 // boolean true if there is data for this field
.OutField6 // boolean true if there is data for this field
.OutField7 // boolean true if there is data for this field
.OutField8 // boolean true if there is data for this field
.OutField9 // boolean true if there is data for this field
.OutField10 // boolean true if there is data for this field
.OutField11 // boolean true if there is data for this field
.OutField12 // boolean true if there is data for this field
.OutField13 // boolean true if there is data for this field
.OutField14 // boolean true if there is data for this field
.OutField15 // boolean true if there is data for this field
```

```
.OutFields // unsigned int that can access all the Field bits at once
.OutGroupGNSS2 // Set of Received Field data for Group7, "GNSS2"
    .OutField0 // boolean true if there is data for this field
    .OutField1 // boolean true if there is data for this field
    .OutField2 // boolean true if there is data for this field
    .OutField3 // boolean true if there is data for this field
    .OutField4 // boolean true if there is data for this field
    .OutField5 // boolean true if there is data for this field
    .OutField6 // boolean true if there is data for this field
    .OutField7 // boolean true if there is data for this field
    .OutField8 // boolean true if there is data for this field
    .OutField9 // boolean true if there is data for this field
    .OutField10 // boolean true if there is data for this field
    .OutField11 // boolean true if there is data for this field
    .OutField12 // boolean true if there is data for this field
    .OutField13 // boolean true if there is data for this field
    .OutField14 // boolean true if there is data for this field
    .OutField15 // boolean true if there is data for this field
    .OutFields // unsigned int that can access all the Field bits at once
.OutGroupX // Set of Received Field data for Group8, Not actually used.
    .OutField0 // boolean true if there is data for this field
    .OutField1 // boolean true if there is data for this field
    .OutField2 // boolean true if there is data for this field
    .OutField3 // boolean true if there is data for this field
    .OutField4 // boolean true if there is data for this field
    .OutField5 // boolean true if there is data for this field
    .OutField6 // boolean true if there is data for this field
    .OutField7 // boolean true if there is data for this field
    .OutField8 // boolean true if there is data for this field
    .OutField9 // boolean true if there is data for this field
    .OutField10 // boolean true if there is data for this field
    .OutField11 // boolean true if there is data for this field
    .OutField12 // boolean true if there is data for this field
    .OutField13 // boolean true if there is data for this field
```

.OutField14 // boolean true if there is data for this field  
.OutField15 // boolean true if there is data for this field  
.OutFields // unsigned int that can access all the Field bits at once  
.Binary\_Message\_CRC // Received message CRC  
.Binary\_Step // Internal to keep track of progress  
.Binary\_Number\_Groups // Internal to keep track of progress  
.Binary\_Number\_Payload // Internal to keep track of progress  
.Binary\_Payload\_Num // Internal to keep track of progress