ADSBee 1090

Open Source Embedded ADS-B Receiver

# Features

* Mode-S Packet Decoding (1090MHz Extended Squitter).
* Multiple output formats over UART or USB:
  + ADSBee CSV
  + MAVLink (not yet implemented)
  + GDL90 (not yet implemented)
* GNSS module input for MLAT or Remote ID applications.
* 2.4GHz 802.11 module for connecting to ADS-B databases via WiFi or broadcasting Remote ID beacon frames in UAS applications (not yet implemented).
* Low profile MHF (U.FL) antenna connector for 1090MHz (ADS-B In) antenna.
* Low profile MHF4 antenna connector for 2.4GHz 802.11 (WiFi) antenna.
* Low Mass and small footprint.
* Integrated M3 mounting holes.
* Firmware updates over USB.

# Applications

* Online ADS-B database feeding.
* Aircraft detection for robotics and embedded projects.
* Remote ID (maybe someday).

# Communication Interface

## UART Interface

The ADSBee 1090 communicates over UART via header pins or its built-in USB interface. Only one interface (USB or UART) is active at a time, and is selected by jumping selector pins XX.

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Baud Rate | 115200 baud |
| Data Bits | 8 |
| Stop Bits | 0 (N) |
| Parity Bits | 1 |
| Logic Level | 3.3V |

## AT Commands

AT Commands are used to configure the ADSBee 1090 receiver’s internal parameters and communication mode.

All AT command arguments are optional. Arguments will be ignored if left as blank or whitespace. For instance, to set the second parameter of AT+MTLSET to 50mV without changing the value of the first parameter, the command “AT+MTLSET=,50” can be sent. Likewise, to change the first parameter to 3mV without changing the value of the second, the command “AT+MTLSET=3,” can be sent.

|  |  |
| --- | --- |
| Command | Parameters |
| **AT+CONFIG**  *Write with echo of value that was set.*  AT+CONFIG=<config:uint16\_t>  +CONFIG=<config:uint16\_t>  *Read present value.*  AT+CONFIG?  +CONFIG=<config:uint16\_t> | **Configuration Command**  config: Config Mode Enable   * 1 = Config mode enabled, printing will be suppressed for configuration AT commands. * 0 = Config mode disabled. Good luck reading anything lol, it’s gonna be wild. |
| **AT+MTLSET**  *Write with echo of values that were set.*  AT+MTLSET=<mtl\_lo\_mv:uint16\_t>,<mtl\_hi\_mv:uint16\_t>  +MTLSET=<mtl\_lo\_mv:uint16\_t>,<mtl\_hi\_mv:uint16\_t>  *Read present set value (stored setpoint, not read by ADC).*  AT+MTLSET?  +MTLSET=<mtl\_lo\_mv:uint16\_t>,<mtl\_hi\_mv:uint16\_t> | **Minimum Trigger Level (MTL) Set Command**  NOTE: mtl\_lo\_mv should be set to a value lower than mtl\_hi\_mv.  Reducing the level of mtl\_lo\_mv will make the receiver more sensitive to weak RF signals, but will also increase the noise that it receives.  Increasing the difference between mtl\_lo\_mv and mtl\_hi\_mv will filter out signals with smaller dynamic range (difference in power level between max amplitude and min amplitude), thereby requiring a higher Signal to Noise ratio for a transponder signal to be decoded. This may reduce the likelihood that the ADSBee tries to decode a transponder signal with invalid bits that will trigger a checksum error.  mtl\_lo\_mv: MTL Low Threshold [milliVolts]   * 0-3300 = Low-side trigger threshold of the comparator circuit on the output of the RF detector. Refer to the AD8313 datasheet and adjustable gain stuff for a conversion from mV (RF detector output signal amplitude) to dBm (RF signal power level in).   mtl\_hi\_mv: MTL High Threshold [milliVolts]   * 0-3300 = High-side trigger threshold of the comparator circuit. |
|  |  |
| **AT+MTLREAD**  *Read with echo of values that were read.*  AT+MTLREAD  +MTLREAD=<mtl\_lo\_mv>,<mtl\_hi\_mv> | **Minimum Trigger Level (MTL) Read Command**  Used an ADC to read the value of mtl\_lo\_mv and mtl\_hi\_mv. Should be roughly in line with the values of mtl\_lo\_mv and mtl\_hi\_mv set in the AT+MTLSET section. |
| **AT+HELP**  AT+HELP  <command>:<command help string>  <command>:<command help string>  <…> | **Help Command**  Prints out a list of available commands and their associated help strings. |
| **AT+RXGAIN**  *Set gain to 100x with echo of gain value that was set.*  AT+RXGAIN=100  +RXGAIN=100  *Read gain value.*  AT+RXGAIN?  +RXGAIN=100  *Test +RXGAIN command.*  +RXGAIN=<gain:uint16\_t> | **Receiver Gain Command**  Adjust the gain of the operational amplifier located after the AD8313 in the receive signal chain. Gain is set as a positive integer value between 1-101.  gain: Receiver Gain [ratio]   * 1-101 = Gain value of operational amplifier operating on AD8313 output. |