ADSBee 1090 Firmware Reference Guide

Notes about how the firmware works and why.

John McNelly

john@pantsforbirds.com

# Inter-Processor SPI Communication

## SPI Packet Definitions

This section details raw packet definitions as can be observed on the SPI bus between the ESP32 and RP2040.

Inter-processor SPI communication is done with maximum transfer lengths of 4096 Bytes. Transfers of large objects like the Settings struct (up to 8kB) are automatically split into multiple smaller transfers.

### Master (RP2040) to Slave (ESP32)

The RP2040 writes to the ESP32 in order to change configuration parameters and pass along transponder packets that it has received. The RP2040 also has the ability to read from the ESP32 in order to verify that changes have been properly executed and to perform watchdog functionality.

#### Master Single Write to Slave

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Transfer 1** | Master Write Packet | | | | | |
| **Byte** | 0 | 1 | 2:3 | 4:5 | 6:(n-2) | (n-1):n |
| **MOSI** | CMD  kWriteToSlave | ADDR | OFFSET | LEN  (unused, since length can be inferred from clocks) | DATA | CRC |
| **MISO** |  |  |  |  |  |  |

#### Master Single Read from Slave

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Transfer 1** | Master Read Request packet | | | | |
| **Byte** | 0 | 1 | 2:3 | 4:5 | 6 |
| **MOSI** | CMD  kReadFromSlave | ADDR | OFFSET | LEN | CRC |
| **MISO** |  |  |  |  |  |

Handshake line goes HI.

|  |  |  |  |
| --- | --- | --- | --- |
| **Transfer 2** | Slave Read Response Packet | | |
| **Byte** | 0 | 1:(n-2) | (n-1):n |
| **MOSI** |  |  |  |
| **MISO** | CMD  kReadFromSlave | DATA | CRC |

Handshake line goes LO.

### Slave (ESP32) to Master (RP2040)

The ESP32 writes to and reads from the RP2040 in order to request settings data (only the RP2040 has access to EEPROM), and to pass along data received from its network connection (e.g. network console commands, firmware updates, etc).

#### Slave Single Write to Master

Handshake line goes HI.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Transfer 1** | Slave Write Packet | | | | | |
| **Byte** | 0 | 1 | 2:3 | 4:5 | 6:(n-2) | (n-1):n |
| **MOSI** |  |  |  |  |  |  |
| **MISO** | CMD  kWriteToMaster | ADDR | OFFSET | LEN | DATA | CRC |

Handshake line goes LO.

#### Slave Single Read from Master

Handshake line goes HI.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Transfer 2** | Slave Read Request Packet | | | | | Master Read Response Packet | | |
| **Byte** | 0 | 1 | 2:3 | 4:5 | 6 | 7 | 8:(n-2) | (n-1):n |
| **MOSI** |  |  |  |  |  | CMD  kDataBlock | DATA | CRC |
| **MISO** | CMD  kReadFromMaster | ADDR | OFFSET | LEN | CRC |  |  |  |

Handshake line goes LO.

### SPI Exchange Examples

Below is a non-exhaustive list of example transactions illustrating how some of the SPI packet definitions are used for communication between the RP2040 and ESP32.

#### RP2040 writes small object to ESP32 without ACK

* RP2040 asserts chip select.
* RP2040 sends single transfer with CMD = kCmdWriteToSlave.
* RP2040 de-asserts chip select.

#### RP2040 writes small object to ESP32 with ACK

* RP2040 to ESP32 single write
  + RP2040 asserts chip select.
  + RP2040 sends single transfer with CMD = kCmdWriteToSlaveRequireAck.
  + RP2040 de-asserts chip select and waits for handshake.
* ESP32 to RP2040 handshake
  + ESP32 asserts HANDSHAKE line.
  + RP2040 asserts chip select.
  + RP2040 reads the first byte of the incoming message to determine that it’s an ACK. If the message is an ACK, the transaction was acknowledged and succeeds, otherwise an error is thrown.
  + RP2040 de-asserts chip select.
  + ESP32 de-asserts HANDSHAKE line.

#### RP2040 writes large object to ESP32 with ACK

* RP2040 breaks large object into multiple single transfers, each of which gets sent in the same manner as “RP2040 writes small object to ESP32 with ACK”. The offset for each transfer gets incremented based on the starting address and size of the previous chunk that was successfully sent.

#### ESP32 reads small object from RP2040 without ACK

* ESP32 asserts HANDSHAKE line.
* RP2040 asserts chip select.
* RP2040 reads first byte of incoming message and determines that it’s a kCmdReadFromMaster. RP2040 reads the address, offset, and length fields and writes its response into the remainder of the packet.
* RP2040 de-asserts chip select.
* ESP32 de-asserts HANDSHAKE line.

## Object Dictionary

Within each SPI packet, object dictionary definitions are used to convert the address, offset, and payload fields into actions that are performed on the RP2040 and ESP32. For instance, the RP2040 can write to the kAddrRawTransponderPacket address via SPI in order to forward a received transponder packet to the ESP32’s onboard aircraft dictionary.