**Serial Communication System**

**Specification**

Author: Anton Kypiatkov

Date: 29/06/2019

**Table of Contents**

1 General 4

2 Main Requirements 4

3 Main system components 4

4 Electrical design 5

5 Power distribution and consumption 7

6 Protocol Description 7

6.1 Serial communication port definitions 7

6.2 Serial communication port protocol 7

6.2.1 TX messages 7

6.2.2 Serial data report message 7

6.2.3 RX messages 8

6.2.4 Setup data message 8

6.2.5 Serial data command message 8

6.3 RF ICD 9

6.3.1 RF TX mess 9

6.3.2 TX RF data message 9

6.3.3 RF RX messages 9

6.3.4 RF RX data message 9

7 Functional Description 10

7.1 System initialization and setup 10

7.2 System reset 10

7.3 Data transfer 11

8 Electrical ICD 12

9 Mechanical ICD 12

10 List of abbreviations 12

**List of Figures**

[Figure 1 – Serial Communication System components 4](#_Toc13428721)

[Figure 2 - Board Connections 5](#_Toc13428722)

[Figure 3 - NRF24L01 pinout 5](#_Toc13428723)

[Figure 4 - Arduino UNO pinout 6](#_Toc13428724)

[Figure 5 - Connection board Schematic 6](#_Toc13428725)

[Figure 6 - Serial comm. system enclosure 12](#_Toc13428726)

# 

**List of Tables**

[Table 1 – Setup ACK message 7](#_Toc12728262)

[Table 2 - Serial data report message 8](#_Toc12728263)

[Table 3 - Setup data message 8](#_Toc12728264)

[Table 4 - Serial data command message 8](#_Toc12728265)

[Table 5 - TX RF data message 9](#_Toc12728266)

[Table 6 - RF RX Data message 10](#_Toc12728267)

# General

The serial communication system connects the quad copter ground control system to the quad copter. It is a RF communication system based on the 2.4GHz band.

# Main Requirements

The following are the top level requirements for the serial communication system:

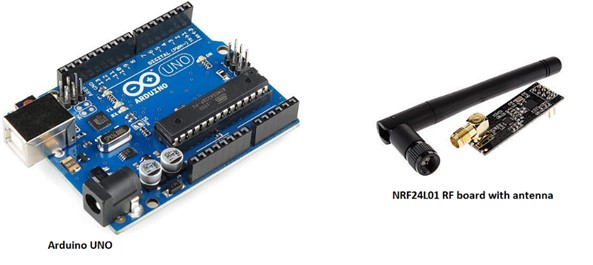
1. The serial communication system shall be connected to the ground controller via a USB port.
2. The serial communication system shall be powered via the USB port.
3. The serial communication system shall communicate with the quad copter using the 2.4GHz RF band.
4. The baud rate for serial communication and the RF port shall be fixed and predefined. For additional information see **section ‎6 – Protocol description.**
5. The serial communication system shall receive message from the quadcopter and forward them to the ground station and vice versa.
6. The following parameters of the communication system shall be configurable (see detailed description in **section ‎5 – System initialization and setup**).
   1. The communication system source address.
   2. The communication system destination address.
   3. The communication system communication channel.
   4. The communication system transmission power.

# Main system components

The system shall be composed of the following components:

1. An Arduino UNO board which shall be used the system controller and implement the serial communication systems software.
2. A NRF24L01 board with a LNA antenna – Which is used to transmit and receive data from the quadcopter.
3. A connection board, which connects the NRF23 transceiver module to the Arduino board.
4. A system enclosure – which houses all other components.

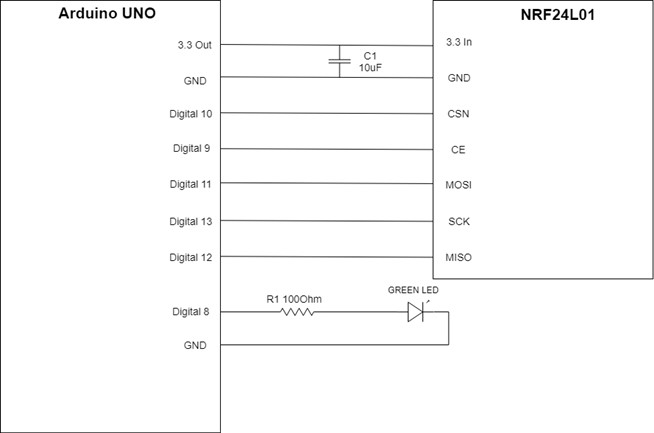
The following figure depicts the system components:



**Figure 1 – Serial Communication System components**

# Electrical design

The following figure depicts the internal connections of the Arduino to the NRF24L01 board:

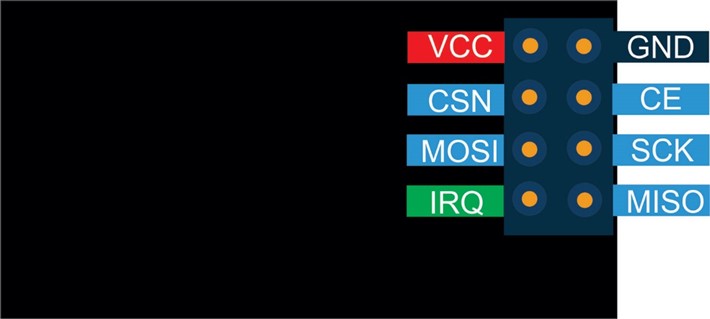


**Figure 2 - Board Connections**

The components values are:

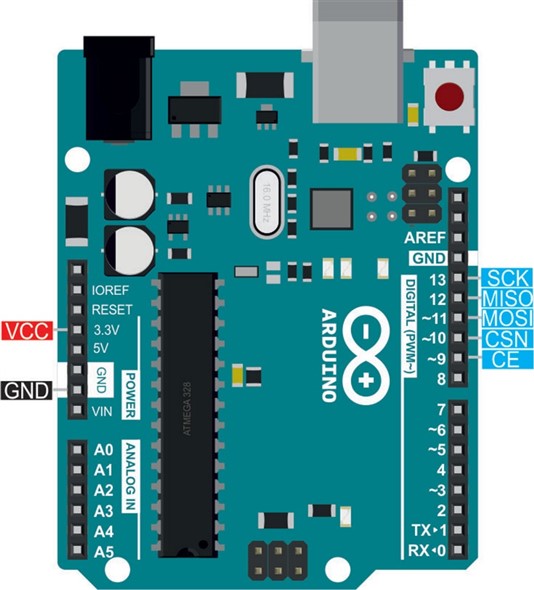
* C1 capacitor value – 10uF
* Resistor Value – 100Ohm resistor 1/8W
* Led Type – Green LED 5mm size. Forward voltage 3.4[V]. Max current 20[mA].

The following diagram depicts the position of the pins on the NRF24L01 module:



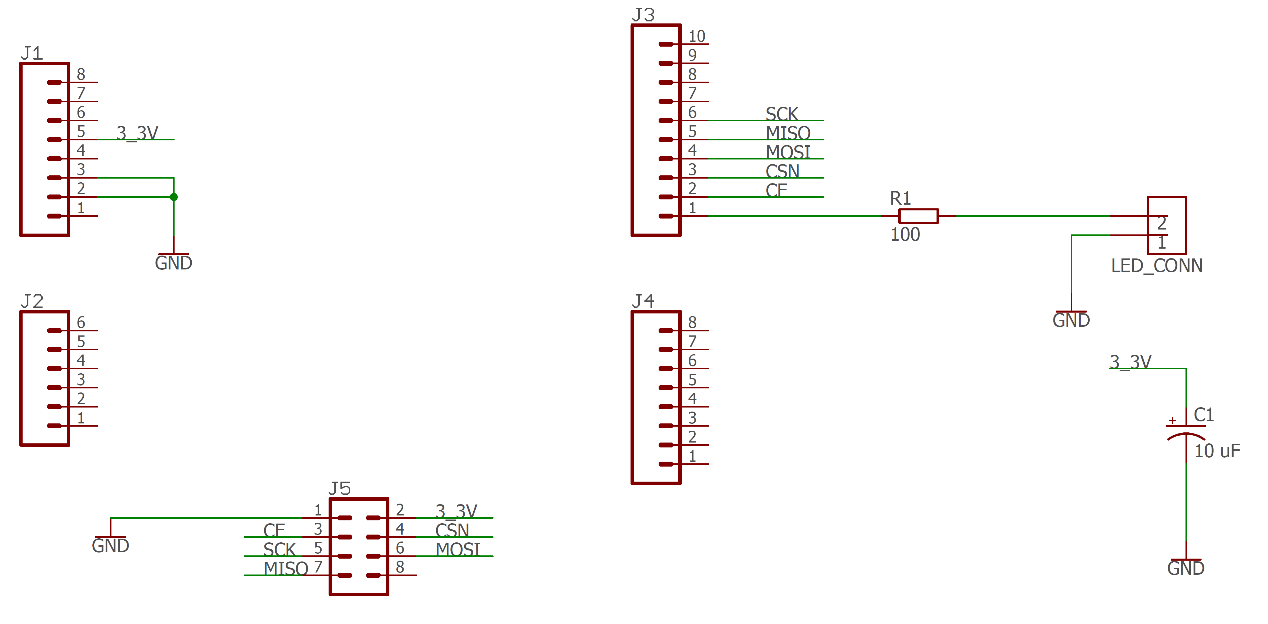
**Figure 3 - NRF24L01 pinout**

The following diagram depicts the location of the pins on the Arduino UNO board:



**Figure 4 - Arduino UNO pinout**

The following is a schematic of the connection board, which connects the Arduino UNO board to the NRF24L01 module:



**Figure 5 - Connection board Schematic**

# Power distribution and consumption

The power input for the serial communication system is from a PC USB 2.0/3.0 port. Which means the system is connected to a 5[V] power input with a maximal current draw of 500[mA], i.e. 2.5[W].

The power draw of the system is as follows:

* Arduino UNO power consumption is: 250[mW].
* Green LED powering: 20[mA]\*3.4[V] = 68[mW].
* NRF24L01 maximal power consumption is 50[mW].

Total power consumption of 368[mW], which is 15% of the total available power from the USB 2.0/3.0 port.

# Protocol Description

This following are the protocol and message definition for the serial communication system:

## Serial communication port definitions

The baud rate for the serial communication port shall be 115200.

The message shall be of 8 bits, with a single stop bit and no parity bits.

## Serial communication port protocol

### TX messages

#### Setup ACK message

The setup ACK messages is sent by the communication setup after it received and processed a setup data command. The structure of the setup ACK message is provided in **Table 1** .

| **Position [Bytes]** | **Field** | **Size [Bytes]** | **Value** | **Description** |
| --- | --- | --- | --- | --- |
| 0 | Begin Transmission field | 1 | 255 (0xFF) | - |
| 1 | Message length | 1 | 3 | - |
| 2 | Message ID | 1 | 254 | 254 Reserved value for setup ACK |
| 3 | Setup\_ack\_status | 1 | 0 – Fail  1 - OK | Describes the initialization status of the communication system. |
| 4 | CRC | 1 | CRC-8 | - |

**Table 1 – Setup ACK message**

### Serial data report message

The data report message is a message which was received from the RF link. The message is forwarded as is to the serial communication interfaces. The structure of the serial data report message is provided in **Table 2**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position [Bytes]** | **Field** | **Size [Bytes]** | **Value** | **Description** |
| 0 | Begin Transmission field | 1 | 255 (0xFF) | - |
| 1 | Message length | 1 | 1 – Minimum message length  30 – Maximum message length | According to the received message. |
| 2 | Message ID | 1 | Any value except for 253/254/255 | 253- 255 are reserved values |
| 3 | Data payload | 0 - 29 | - | Length according to message length – 1 |

**Table 2 - Serial data report message**

### RX messages

### Setup data message

The setup data message to the communication system during initialization in order to initialize the parameters of the RF link. The structure of the setup data message is provided in **Table 3**.

| **Position [Bytes]** | **Field** | **Size [Bytes]** | **Value** | **Description** |
| --- | --- | --- | --- | --- |
| 0 | Begin Transmission field | 1 | 255 (0xFF) | - |
| 1 | Message length | 1 | 6 | - |
| 2 | Message ID | 1 | 253 | 253 Reserved value for setup message ID |
| 3 | Tx\_address | 1 | 0 - 255 | - |
| 4 | Rx\_address | 1 | 0 - 255 | - |
| 5 | Selected\_RF\_channel | 1 | 0 - 125 | For frequency range of:  2.400 to 2.525 [GHz] |
| 6 | Power | 1 | 0 – 3 | 0 for -18dBm, 1 for -12dBm, 2 for -6dBm and 3 for 0dBm |
| 7 | CRC | 1 | CRC-8 | - |

**Table 3 - Setup data message**

### Serial data command message

The serial data command message is a message received over the serial port which is required to be forwarded to the RF channel. The structure of the serial data command message is provided in **Table 4**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position [Bytes]** | **Field** | **Size [Bytes]** | **Value** | **Description** |
| 0 | Begin Transmission field | 1 | 255 (0xFF) | - |
| 1 | Message length | 1 | 1 – Minimum message length  30 – Maximum message length | According to the received message. |
| 2 | Message ID | 1 | Any value except for 253/254/255 | 253- 255 are reserved values |
| 3 | Data payload | 0 - 29 | - | Length according to message length - 1 |

**Table 4 - Serial data command message**

## RF ICD

### RF TX mess

### TX RF data message

The RF data message is received from the serial channel and forwarded over the RF link with the same content and structure. The structure of the TX RF data message is provided in **Table 5**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position [Bytes]** | **Field** | **Size [Bytes]** | **Value** | **Description** |
| 0 | Begin Transmission field | 1 | 255 (0xFF) | - |
| 1 | Message length | 1 | 1 – Minimum message length  30 – Maximum message length | According to the received message. |
| 2 | Message ID | 1 | Any value except for 253/254/255 | 253- 255 are reserved values |
| 3 | Data payload | 0 - 29 | - | Length according to message length - 1 |

**Table 5 - TX RF data message**

### RF RX messages

### RF RX data message

The RF RX data message is received from the RF channel and is to be forwarded as is to the serial channel. The structure of the message is provided in **Table 6**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position [Bytes]** | **Field** | **Size [Bytes]** | **Value** | **Description** |
| 0 | Begin Transmission field | 1 | 255 (0xFF) | - |
| 1 | Message length | 1 | 1 – Minimum message length  30 – Maximum message length | According to the received message. |
| 2 | Message ID | 1 | Any value except for 253/254/255 | 253- 255 are reserved values |
| 3 | Data payload | 0 - 29 | - | Length according to message length - 1 |

**Table 6 - RF RX Data message**

# Functional Description

This section describes the functional requirements for the communication system.

## System initialization and setup

When the system is first powered on it shall wait to receive a setup message from the controller, via the USB connection, with the following parameters:

* TX address (the source address of the serial communication system) – Range of values is 0 – 255.
* RX address (the destination address of the quad copter) – Range of values is 0 – 255.
* Selected channel – defines the frequency channel to be used – range 0 -125 for frequencies ranging between 2.400 to 2.525 [GHz].
* The transmission power – values can be: 0 for -18dBm, 1 for -12dBm, 2 for -6dBm and 3 for 0dBm.

Once the message is received and the values are defined the serial communication system shall open the RF RX/TX port with the received parameters, and shall acknowledge to the controller, via the USB port, that the setup data was received. This shall be done by sending a setup ACK message over the serial USB port.

Additionally, the communication system turns on the led on Arduino pin 8 after power up or reset.

## System reset

Due to the software package implementation of the Arduino board , the Arduino boards boot loader resets each time the serial communication port is opened. For our application of the serial communication system this is a welcomed behavior.

Each time a new serial connection is established via the serial USB communication port, the setup and initialization of the Arduino board must be re-run. Hence after a system reset the Arduino goes back to waiting for a setup message to arrive.

## Data transfer

This section describes the data transfer algorithm used by the communication system. The communication system receives data and sends data simultaneously on both physical ports: a USB serial port and an RF 2.4GHZ port.

The following is the data transfer algorithm for the serial communication system:

1. Open serial port with the following properties: 115,200 baud rate, 8 bits, 1 stop bit, no parity.
2. Wait for a setup message.
   1. If a correct setup message arrived:
      1. Reply with an acknowledge message.
      2. Open the RF port according to the definitions in the setup message.
   2. Else reply with an error code Acknowledge message and go back to waiting for a setup message.
3. Once a setup message was received start reading from both the serial and RF ports in sequence as follows:
   1. Read a single byte from the serial port.
      1. If the byte is a msg\_header start waiting for message length to arrive, else continue.
      2. If message header was read,read message length and continue
      3. If a message header and length was read, read a byte of data and store it in a message buffer.
      4. If all bytes were read, send an RF message with the data copied from the message buffer and go back waiting to a message header.

**Note:** Between read bytes check if there was no timeot, i.e. time between read bytes > Thresh\_serial\_wait.

* 1. Read a single byte from the RF port.
     1. If the byte is a msg\_header start waiting for message length to arrive, else continue.
     2. If message header was read,read message length and continue
     3. If a message header and length was read, read a byte of data and store it in a message buffer.
     4. If all bytes were read, send a message over the serial USB port with the data copied from the message buffer and go back waiting to a message header.

**Note:** Between read bytes check if there was no timeot, i.e. time between read bytes > Thresh\_RF\_wait.

# Electrical ICD

The physical inputs and outputs of the system are based on the Arduino UNO’s DC input and the input USB port.

**Note:** The DC input should be used only when the USB port is physically disconnected from the serial communication system.

The pin-out of the serial communication system is provided in table ###

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector** | **Pin #** | **Function** | **Notes** |
| USB TYPE B | 1 | Vcc | +5VDC |
| USB TYPE B | 2 | Data - |  |
| USB TYPE B | 3 | Data + |  |
| USB TYPE B | 4 | GND |  |
| DC JACK | In | VDC | 6VDC – 20VDC |
| DC Jack | Out | GND |  |

**Table 7 - System pinout**

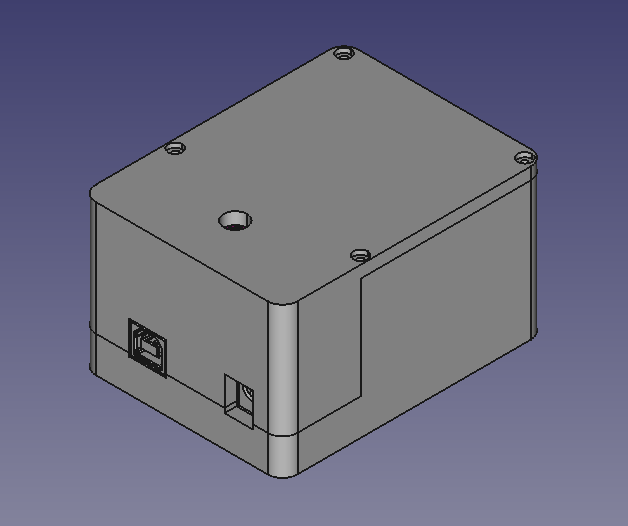
# Mechanical ICD

This section provides a description of the mechanical properties of the system.

The weight of the system, without an attached USB cable is less than 0.15[Kg].

The system enclosure is composed of PLA plastic, see **Figure 6** for a depiction of the enclosure.

The dimensions of the system are: width 67.3[mm], height 50[mm], length 87.8[mm].



**Figure 6 - Serial comm. system enclosure**

# List of abbreviations

TBD