

J4210 Arduino Interfacing

Contents

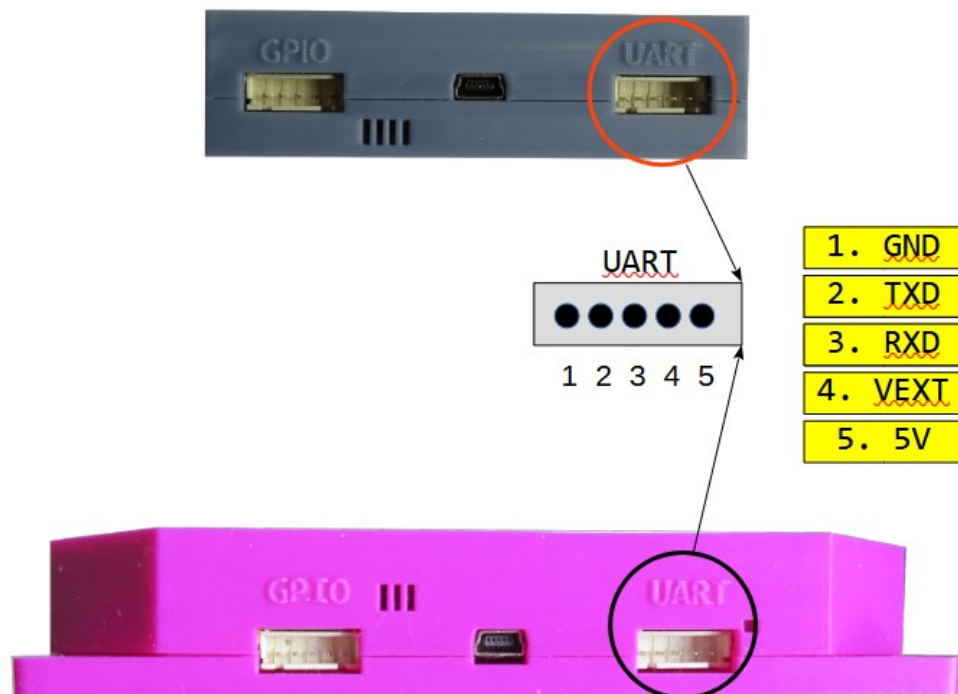
Getting Started.....	3
Hardware Description:.....	3
Hardware Setup:.....	4
Connection basic.....	4
Power Requirements.....	4
Connection Schemes for different Microcontrollers:.....	4
For Arduino Mega:.....	5
FOR ESP32 dev module (wroom):.....	5
For Arduino Leonardo.....	6
For Arduino Nano esp32.....	6
For RPI Pico.....	7
Software Installation:.....	7
Common Troubleshooting:.....	8

Getting Started

This documentation is your comprehensive guide to understand and utilize the full potential of **Jence Uhf RFID Reader**. Whether you are a developer, system integrator, or end-user, we have tailored this guide to assist you in setting up, configuring, and integrating our UHF RFID reader into your specific applications.

Hardware Description:

Aside from USB interface our reader provides UART communication interface via Rx and Tx pin. The configuration picture is given below

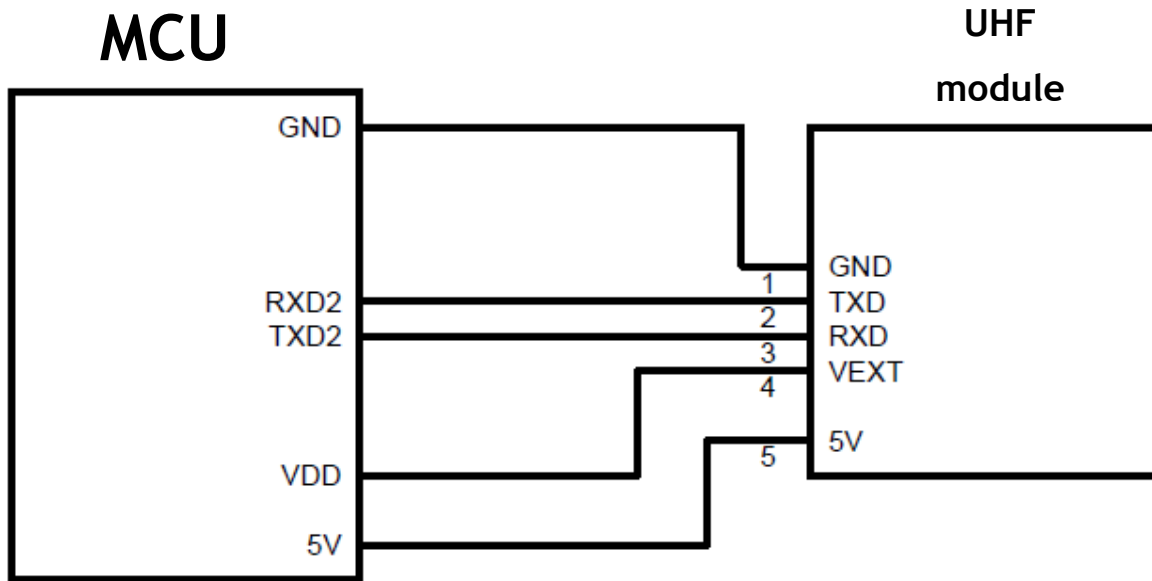


UART - RXD and TXD will be crossed with the MCU's TXD and RXD.

VEXT - This is the Logic High, VOH voltage. In 3.3V logic, tie it to 3.3V, in 5V logic, tie it to 5V.

5V - This is supply voltage. Do not supply 3.3V here.

The basic idea is to connect the uhf reader module with an MCU like below configuration



Hardware Setup:

Connection basic

The key is MCU needs 2 sets of UART port. One is to connect with the PC through USB-Serial device to monitor the status of the MCU through Serial Monitor. Another set of UART is to communicate with the module to send command and retrieve information. In the example, we are using UART2 for the Module - MCU communication but it is configurable as the user wants. Only 1 set of UART pins can do the communication if Serial Monitor is not needed. Our example uses 2 sets of UART port.

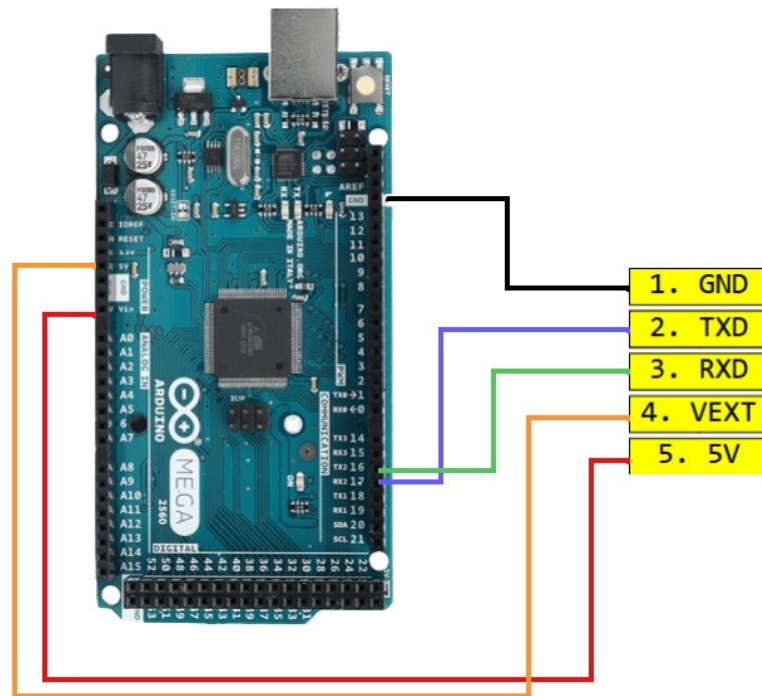
Power Requirements

UHF RFID Module requires 400mA, 5V power while scanning. If the MCU is used as power source for the module it will take around 400mA of current while scanning. While scanning if the MCU can't provide that much current then scan will fail. So, if the MCU can't provide 400mA 5V then external power source is needed.

Connection Schemes for different Microcontrollers:

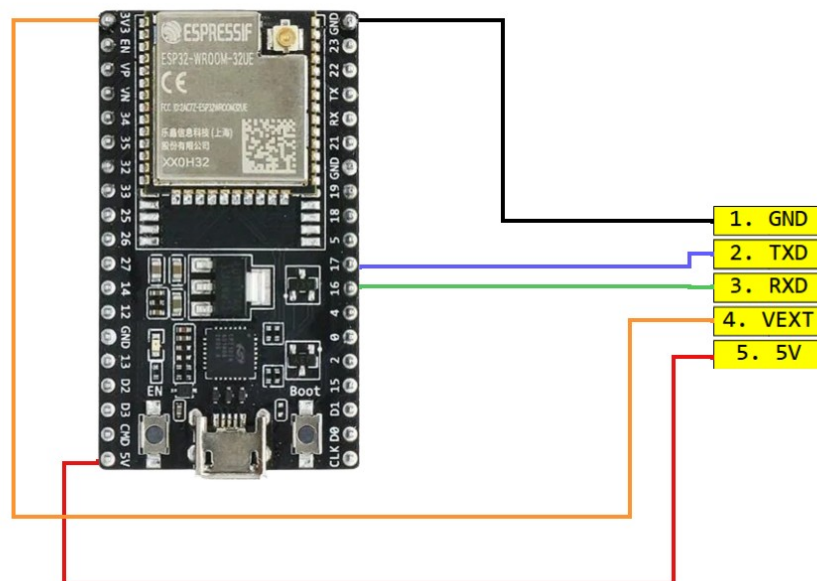
Connection configuration example for some common microcontroller that can be used with Arduino platform is provided below. Note that pin configuration is just an example and an expert user can configure their pins however they want reflecting their code.

For Arduino Mega:

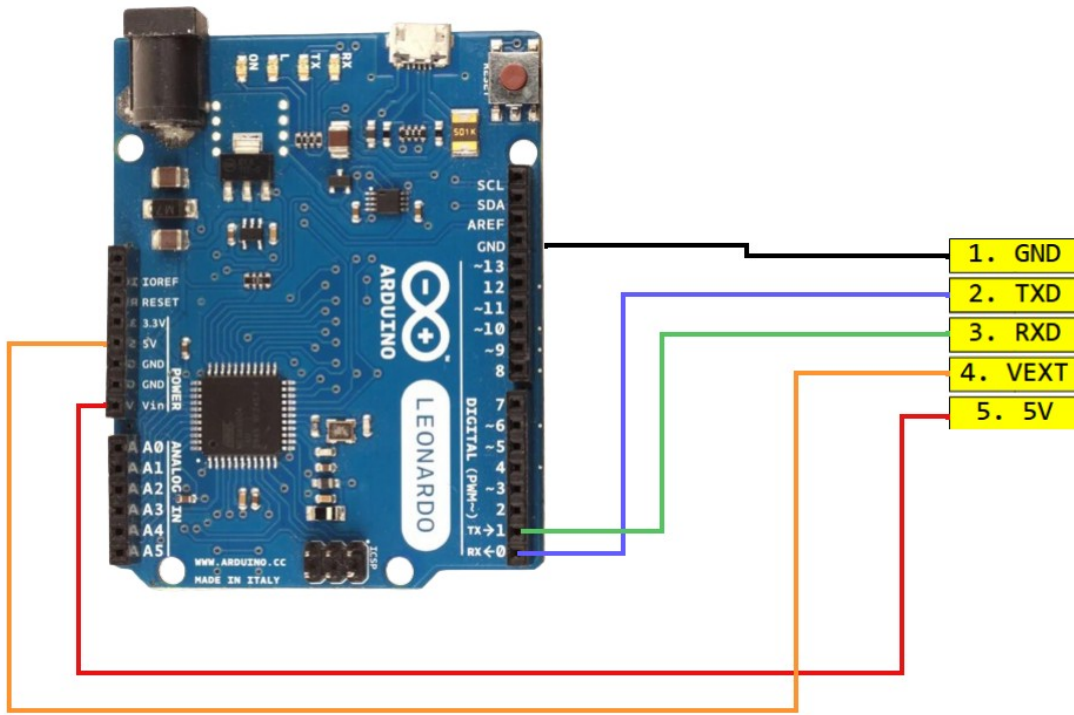


f

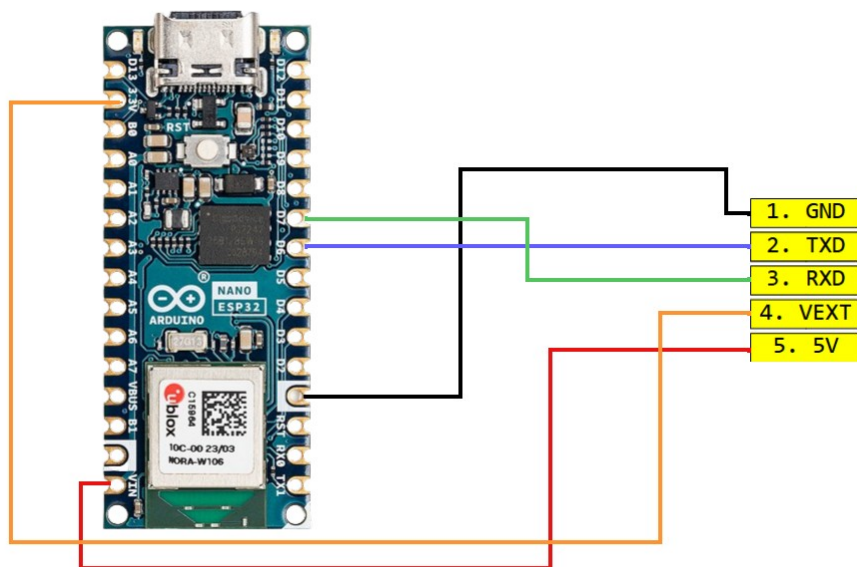
FOR ESP32 dev module (wroom):



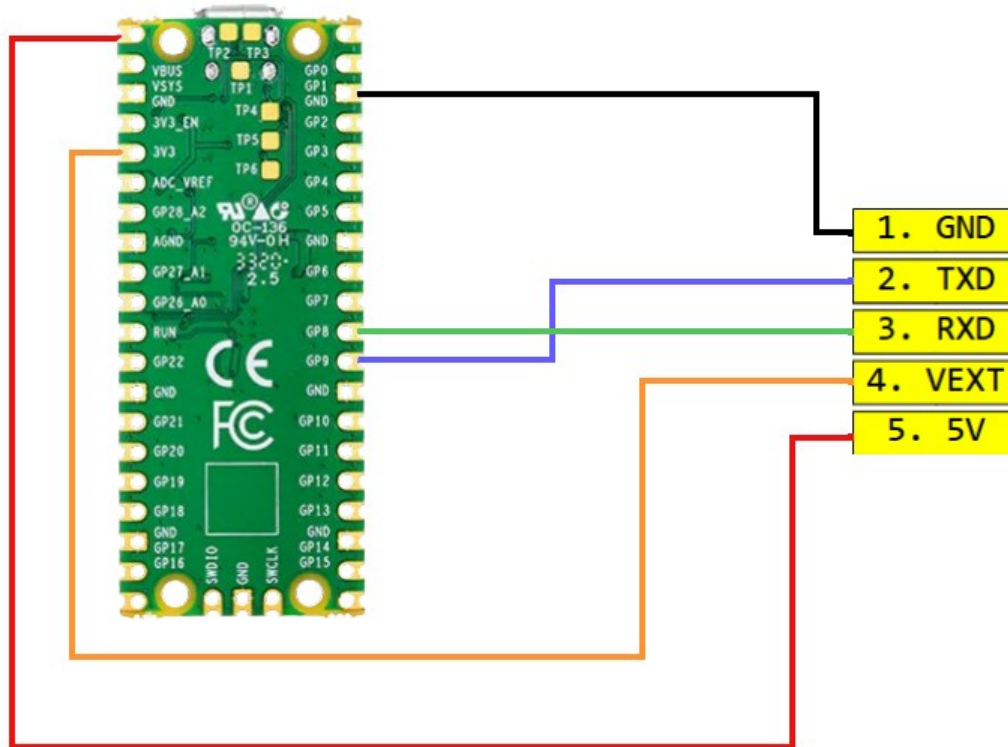
For Arduino Leonardo



For Arduino Nano esp32



For RPI Pico



Software Installation:

UHF RFID readers has an Arduino framework library class named J4210U. In the J4210U.h file the public methods are to be used to communicate with the RFID reader. For the library class, an example sketch is also provided that does some operation to demonstrate some of the functionality of the library.

Software setup:

1. Extract the zip file
2. Open j4210u-driver-embedded.ino file with Arduino IDE, it contains the example for J4210U library class.
3. Check if J4210U.h and J4210U.cpp file has been loaded with the .ino file. If not then manually add them with Arduino IDE.
4. Connect the wires accordingly based on the MCU

5. Build and upload the sketch setting the right configuration for the MCU with TOOLS tab in Arduino IDE
6. Check in serial monitor for messages. The default Serial monitor baud of the example is 9600
7. A successful integration will show the reader settings in the serial monitor and the reader will keep try to scan nearby cards after 3s. keep some uhf tag nearby and EPC and TID will be visible after each scan in the serial monitor.
8. After testing the example you can start using the library for Standard application such as GetSettings, SetSettings, Inventory Scan, Read, Write, GPIO, SetPassword, Lock, Kill, , operation etc. by using the public methods of J4210U class

Common Troubleshooting:

1. Check the device if it behaves intendedly with platform specific [desktop software](#) before using it with MCU.
2. While connecting the wires sometimes Rx and Tx might get mixed up by mistake and that can cause failed communication. Check if the **Rx** and **Tx** is connected properly and try swapping the Rx and Tx.
3. VEXT is the logic voltage. If VEXT doesn't get matched input voltage as the logic HIGH level then the communication won't work. Check the voltage level of VEXT
4. Module is connected but scan can't find any tags. It might be due to the module is not getting enough power to scan. Connect an external power source with 5V and GND pin.
5. Check the baud Rate from desktop software to be 100% sure. Default baud rate and Arduino example has also set to 57600bps. If that baud rate fails then try changing it to 115200bps.