RS-485 Interfacing through Odroid-C2 and Arduino

Alok Ranjan Kesari[†] and G V V Sharma[*]

Contents

1 Components

Component	**Quantity**
USB-RS485	1
Max485	2
Arduino	2
Known Resistors	2
Unknown Resistors	2

TABLE 1

2 Single Slave

2.1 Hardware Connections

Problem 2.1. Make the pin connections as in Table 2. Also connect the Resistors R_1 and R_2 according to Fig. 2.1.

2.2 Software Setup

For Arduino: This library needs to be copied to the libraries folder in the sketchbook directory of Arduino. https://github.com/smarmengol/Modbus-Master-Slave-for-Arduino

† The author wan an intern at IIT Hyderabad during the summer of 2017 email: alok.kesari@yahoo.co.in. *The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

Arduino	**Max485**	**USB-RS485**
Tx	DI	
Rx	RO	
D4	DE	
D4	RE	
5V	VCC	
GND	GND	
	A	A
	B	B

TABLE 2

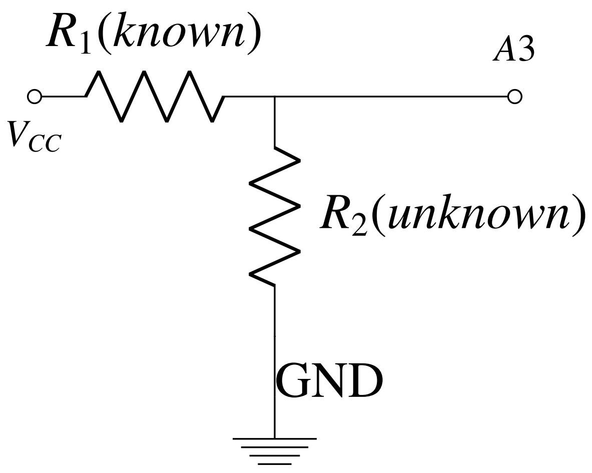

Fig. 2.1

2.3 Configuring Arduino as slave

Problem 2.2. Run the following program on the odroid using Arduino software. This will configure the Arduino as a slave.

```
#include "Arduino.h"
#include <ModbusRtu.h>
#define TXEN  4

//Storing resistance value
uint16_t resistance[1] = {0};
```

```
//To initialize the slave arduino
   with address 1
//And using 0/1 pin of arduino as
   TX/RX
//TXEN enables MAX485 transmission
Modbus slave(1,0,TXEN);

void setup() {
//analog pin A3 used for received
   resistance value
  pinMode(A3, INPUT);
 //Baudrate at which modbus works
  slave.begin(19200);
}
void loop() {
  resistance[0]=(analogRead(A3));
  slave.poll(resistance, 1);
}
```

2.4 On Odroid

The following commands are for an Odroid running Archlinuxarm. The process for running it on other Linux distributions is similar.

```
#For minimalmodbus,
#python module to control RS-485
sudo pip install minimalmodbus
```

2.5 Odroid as Master

Problem 2.3. Run the following code on Odroid. You will see the resistance value being displayed.

```
#!usr/bin/env python

import time
import minimalmodbus
import serial


instrument = minimalmodbus.
   Instrument('/dev/ttyUSB0',1)
vi=5;#VCC
r1=1000;#1K known resistance

#Modbus Configuration for Master
instrument.serial.baudrate   =
   19200
instrument.serial.bytesize   = 8
instrument.serial.parity     =
   serial.PARITY_NONE
instrument.serial.stopbits   = 1
instrument.serial.timeout    = 1
instrument.mode              =
   minimalmodbus.MODE_RTU


while 1:

        try:
                #Reading resistor
                   [0] in slave
                test_reg =
                   instrument.
                   read_registers
                   (0,1)
        #
            print (test_reg)
                #Calculating the
                   resistance using
                    voltage level
                vo=(test_reg[0]*vi
                   )/1024.0;
                b=(vi/vo)-1;
                r2=r1/b;
                print ('The
                   resistance value
                    measured from 1
                    is:', r2)
                #polling every 0.5
                    seconds
                time.sleep (0.5)
        except:
                print ("error USB2
                    -----")
                time.sleep (1)
```

3 Multiple Slaves

Problem 3.1. Configure another arduino as a slave with address 2. Use problem 2.2.

Problem 3.2. Connect one more arduino to the odroid using a breadboard according to Table 2. You will have to make multiple connections using the breadboard.

Problem 3.3. Modify the code in problem 2.3 to verify if the second arduino is active.

Problem 3.4. Run the following program to control both arduinos on the RS-485 bus. Note that the bus is nothing but the connection from multiple RS-485 interfaces on the common A,B lines.

```
#!usr/bin/env python

import time
import minimalmodbus
import serial


instrument = minimalmodbus.
   Instrument('/dev/ttyUSB0',1)
instrument2 = minimalmodbus.
   Instrument('/dev/ttyUSB0',2)
vi=5;
b=0;
r1=1000;

instrument.serial.baudrate   =
   19200
instrument.serial.bytesize   = 8
instrument.serial.parity     =
   serial.PARITY_NONE
instrument.serial.stopbits   = 1
instrument.serial.timeout    = 1
instrument.mode              =
   minimalmodbus.MODE_RTU

instrument2.serial.baudrate  =
   19200
instrument2.serial.bytesize  = 8
instrument2.serial.parity    =
   serial.PARITY_NONE
instrument2.serial.stopbits  = 1
instrument2.serial.timeout   = 1
instrument2.mode             =
   minimalmodbus.MODE_RTU

usb1_on = True
usb2_on = True

while 1:
        if usb2_on == True :
                try:
                        print ("
                           USB1")
                        test_reg =

                           instrument2
                           .
                           read_registers
                           (0,1)
                        print (
                           test_reg
                           [0])
                        vo=(
                           test_reg
                           [0]*vi)
                           /1024.0;
                        b=(vi/vo)
                           -1;
                        r2=r1/b;
                        print ('
                           The␣
                           resistance
                           ␣value␣
                           measured
                           ␣from␣1␣
                           is:', r2
                           )
                        time.sleep
                            (0.05)
                except:
                        print ("
                           error␣
                           USB1")
                        time.sleep
                            (1)


        if usb1_on == True :
                try:
                        print ("
                           USB2")
                        test_reg =

                           instrument
                           .
                           read_registers
                           (0,1)
                        print (
                           test_reg
                           )
                        vo=(
                           test_reg
                           [0]*vi)
                           /1024.0;
                        b=(vi/vo)
```

```
                    −1;
                r2=r1/b;
                print ('
                    The␣
                    resistance
                    ␣value␣
                    measured
                    ␣from␣2␣
                    is:', r2
                    )
                time.sleep
                    (0.5)
            except:
                print ("
                    error␣
                    USB2␣")
                time.sleep
                    (1)
```