**DATA COMMUNICATION**

**LAB 8: SPI RFID READER**

# Introduction to RFID

Radio-Frequency Identification (RFID) is the use of radio waves to read and capture information stored on a tag attached to an object.  A tag can be read from up to several feet away and does not need to be within direct line-of-sight of the reader to be tracked.

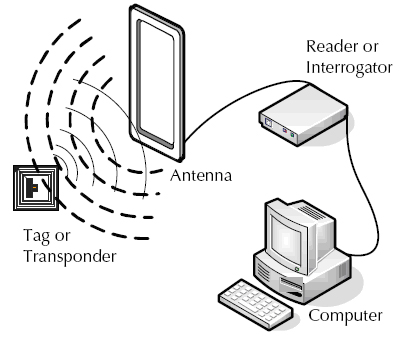


Figure 1 FRID

There are two types of RFID tags: passive and battery powered.  A passive RFID tag will use the interrogator’s radio wave energy to relay its stored information back to the interrogator.  A batter powered RFID tag is embedded with a small battery that powers the relay of information.

The MFRC522 this the RFID reader. It works with the frequency of 13.56MHz.

# Getting started with MFRC522

# Firstly, download the Arduino library for MFRC522 from this link: <https://github.com/miguelbalboa/rfid>

# 

Figure 2: Download RFID MFRC522 Library

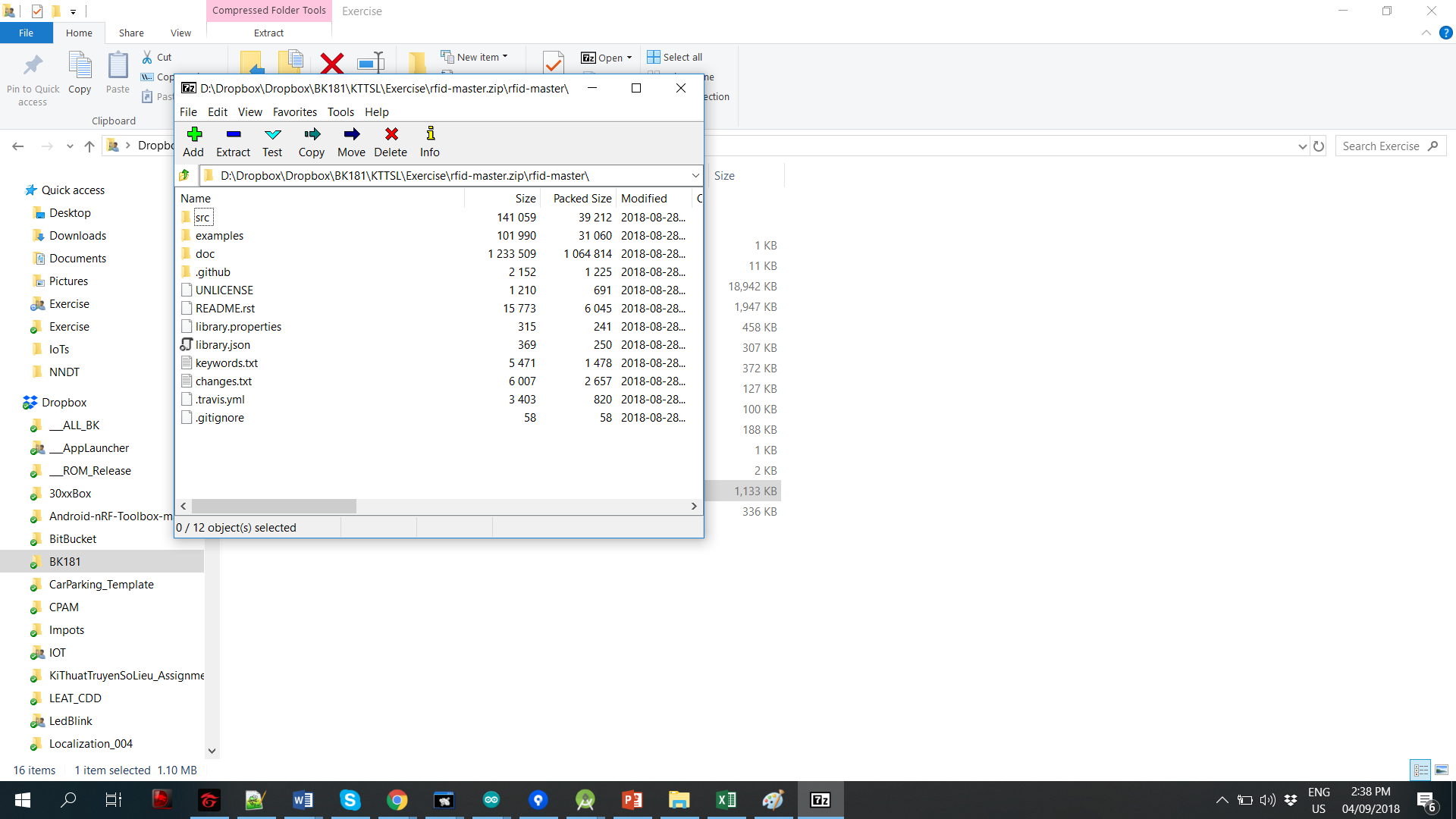


Figure 3: Unzip the download file

After unzip the file (see Figure 3), copy the whole folder **rfid-master** to the Arduino library folder (normally located at **This PC**\**Documents\Arduino\libraries**), as shown in Figure 4. Finally, rename this folder to **rfid**

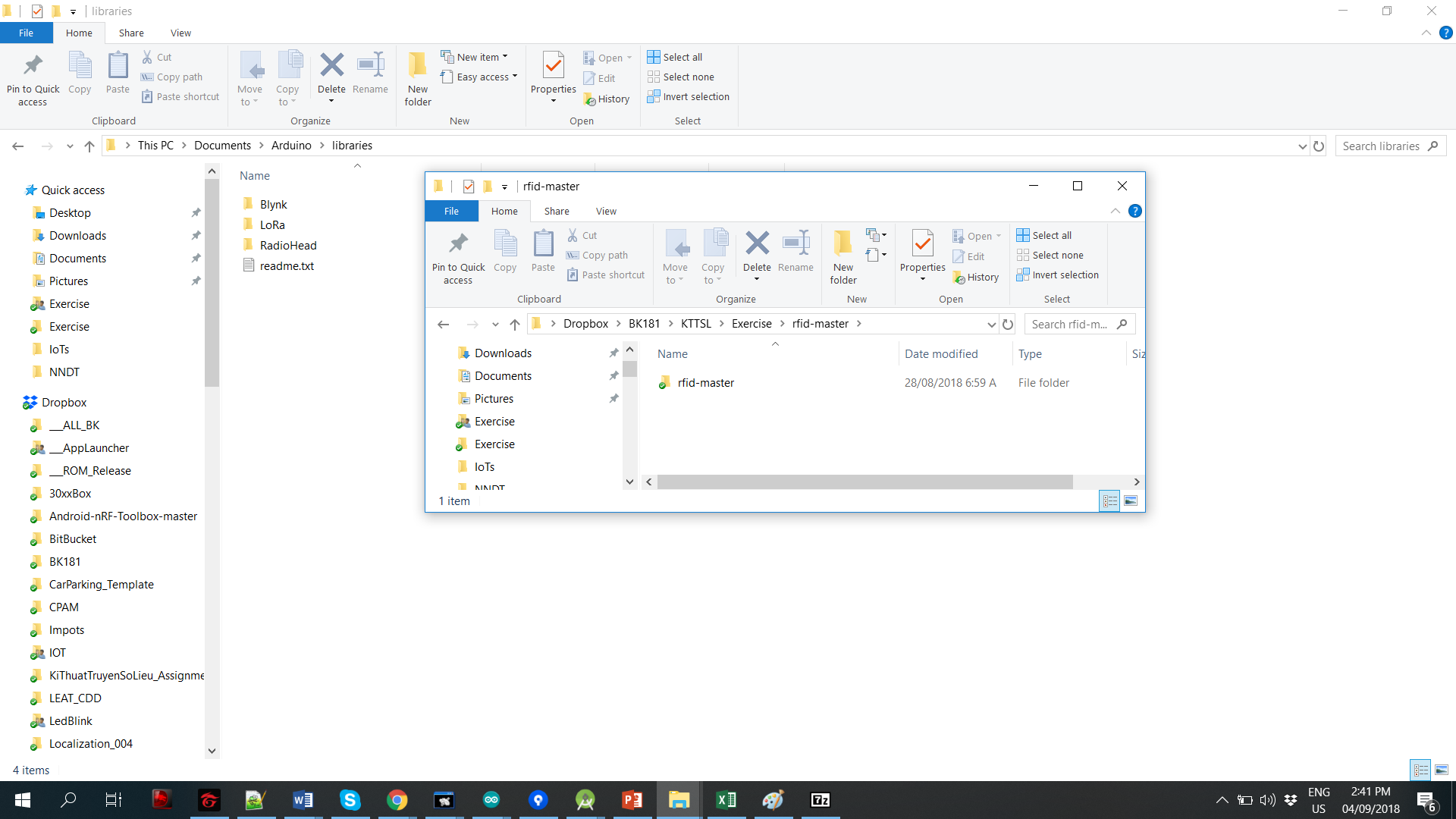


Figure 4: Copy the library to Arduino library folder

# Next step, close the Arduino IDE if it is open then, restart the IDE again. Navigate to the first example located at File/Examples/MFRC522/DumpInfo (see Figure 5)

# 

Figure 5: RFID Example with DumpInfo

# Connect the MFRC522 with the NodeMCU like the following figure:

# Picture of Pin Wiring

# 

Figure 6 SPI connection between NodeMCU and MFRC522

Finally, upload the example code to the Node MCP. Then, open the serial monitor, you should see something like the image below.

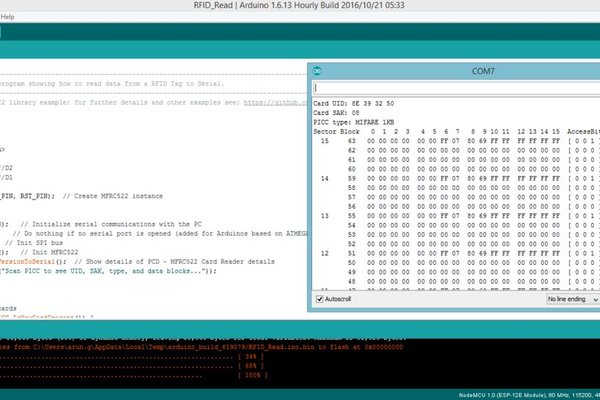


Figure 7 Open serial monitor

Note your UID card ("Card UID : xx xx xx xx ") as it will be used later in the exercise part.

# Exercise

Implement a short program to turn on the LED only if your card is presented. The LED will be turn on for 3s and then OFF. Design your system using DFA and then implement the code according to your design.

*/\* Before reading the code, please take a time reading my comment: I followed the instructions listed here and it did not work, not even a pinout. I do not know whether you’ve tested the pinout and the code, I think yes because of the pictures above were captured at the beginning of this semester. You might understand that debugging hardware components is much more complicated compared to software. I have even soldered the circuit 3 times and finally, the problem was due to the code, the Arduino board and the pinout. If you want to make it easier for later courses, please fix your instruction\**/

switch (state){

case WAITING:{

if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial())

state = VERIFY;

break;

}

case VERIFY:{

for (byte i = 0; i < mfrc522.uid.size; i++) {

if (mfrc522.uid.uidByte[i] != CARD1[i]){

state = WAITING;

Serial.println("Access Denied");

return;

}

}

Serial.println("Access Accepted");

state = LED;

digitalWrite(LED\_BUILTIN, LOW);

timePrev = millis();

break;

}

case LED:{

timeCurrent = millis();

if (timeCurrent - timePrev >= 3000){

digitalWrite(LED\_BUILTIN, HIGH);

state = WAITING;

}

break;

}

}

# Extra exercise

1. As you have learned, the SPI allows one master to connect to many slaves, now try to connect 2 MFRC522 with one NodeMCU. Reference at <https://www.youtube.com/watch?v=HmGmFknAIqc>
2. Assumed that your data: Name, age, student ID…are written in the RFID tag. Each time when you tap your tag, the MFRC522 will read your data then the NodeMCU will upload these data to google sheet. Write a program for your NodeMCU. References are listed below:

<https://www.youtube.com/watch?v=L68MxmHewww>

<https://www.hackster.io/detox/transmit-esp8266-data-to-google-sheets-8fc617>

<http://embedded-lab.com/blog/post-data-google-sheets-using-esp8266/>