HermesTrack Arduino – RC522 RFID Sensor Interface

Purpose: The purpose of this documentation is to provide a first time user with the tools necessary to implement the open source HermesTrack system for their individual applications.

What it does: The HermesTrack system is an array real time RFID sensors that manages and compiles personnel or product movement data. One example of an application is workflow tracking in hospitals and nursing homes to reduce infection rates and undesirable patient outcomes.

Supplies:

1. Arduino UNO R3 Microcontroller (1)
2. Type A/B male-male USB cable (1)
3. Mifare RC522 RFID Reader/Writer (1)
4. Electrical wiring (10 four inch wires)
5. RFID chip card (2-3)

Assembly Procedure:

1. Remove products from the packaging and line up next to each other
2. Connect the RFID sensor pins with their respective Arduino input pins using the table below as a guide. The pins will be clearly labelled on each device.

Table 1: Arduino to RC522 Sensor Pin Connections

Arduino Uno	RC522 RFID Sensor
Pin 10	SDA
Pin 13	SCK
Pin 11	MOSI
Pin 12	MISO
NC	IRQ
GND	GND
Pin 9	RST
3.3V	3.3V

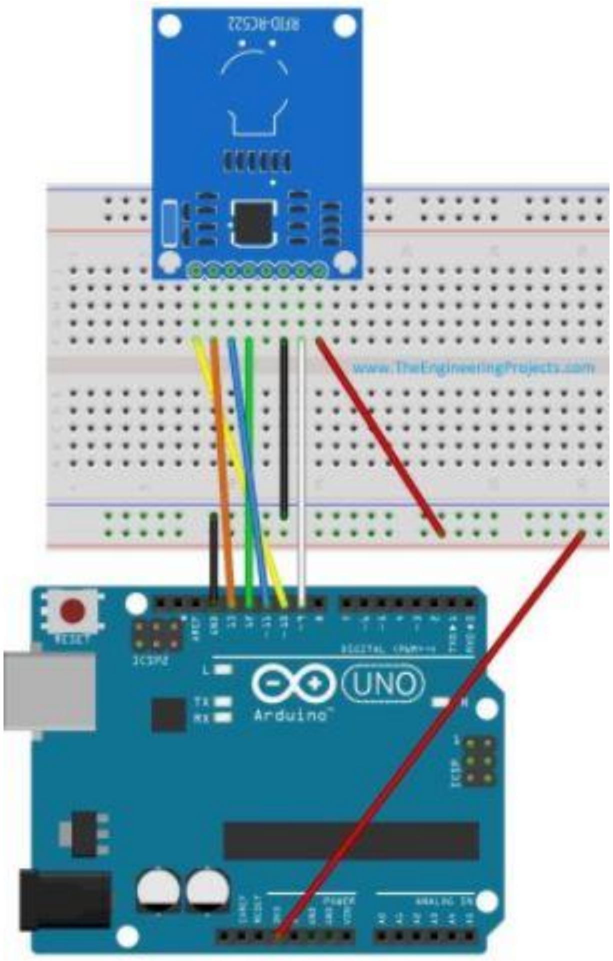

Figure 1: Arduino to RC522 Sensor pinout diagram (www.TheEngineeringProjects.com)

3. On your computer, download the SPI.h and MFRC522.h Arduino libraries in order to have all of the RC522 functions in your Arduino library.
    a. These libraries can be found by searching their name on Google and downloading from an open source website such as Github.
4. Plug the Arduino into the computer and upload the following code.
    a. No modification necessary with 1 RFID sensor per board. However, attempting to use more than 1 sensor per Arduino will require a significant amount of extra work editing the libraries downloaded.
    b. Modifications will need to be made to the code based on your system (Number of sensors, sensor labelling, Arduino to Pi data transfer method)

c.

```
/*
*
*Get identifier information from RFID-RC522 sensors
*Can only connect one RFID sensor per arduino (uses all digital pins)
*The serial output is read into a Raspberry Pi for processing and uploadin
*into cloud; however can also implement using arduino wifi shield.
*
*
*can integrate other sensors within the same arduino, but include under
*if(!mfrc22.PICC_IsNewCardPresent()) so that it is read every loop
*set the same delay inside if statement as end of void loop()
*
*
*/
#include <SPI.h>
#include <MFRC522.h> //library for sensor

#define SS_PIN 10 //SS input pin autodefined in MFRC522
#define RST_PIN 9 //define reset pin
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.

void setup()
{
  Serial.begin(9600);   // Initiate a serial communication
  SPI.begin();      // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
}
void loop()
{
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent())
  { //include input reading for other sensors here
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Show UID on serial monitor
  Serial.print("2 "); //This printed sensor label can be changed to
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
    //sends output in format compatible with raspberri pi processing code
  }
  Serial.println();
 delay(1000);
}
```

5. Now that the code is uploaded, you are able to scan the RFID enabled card or fob and it’s tracking information will print on the Arduino.
    a. Repeat this process for as many RFID scanners you would like to implement into your system.
6. Once hooked up to the Raspberry Pi, the Arduino will send the data information to the Pi through a serial output.
    a. The Raspberry Pi code is more complex, but can be found on Github for upload to the Pi. The code can be found by searching for HermesTrack on Github.

Key Definitions:

- Raspberry Pi: Low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that can be coded in Python or Scratch.
- Arduino: A physical programmable circuit board (microcontroller) and piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload open source computer code to the physical board.
- SDA: Serial Data Line. Communication line that transfers information one bit at a time from the RFID sensor to the Arduino

- SCK: Serial Clock. This line is the output of the clock generator in the Arduino master that dictates the rate of serial data transfer between the devices.
- MOSI: Master Out Slave In. Output command signal from the Arduino master to the RC522 slave device.
- MISO: Master In Slave Out. Data output from the RC522 RFID sensor slave device to the Arduino master.
- SPI: Serial Peripheral Interface. Common communication method used for two-way communication between two devices. Typically this set up consists of a master device and slave device.