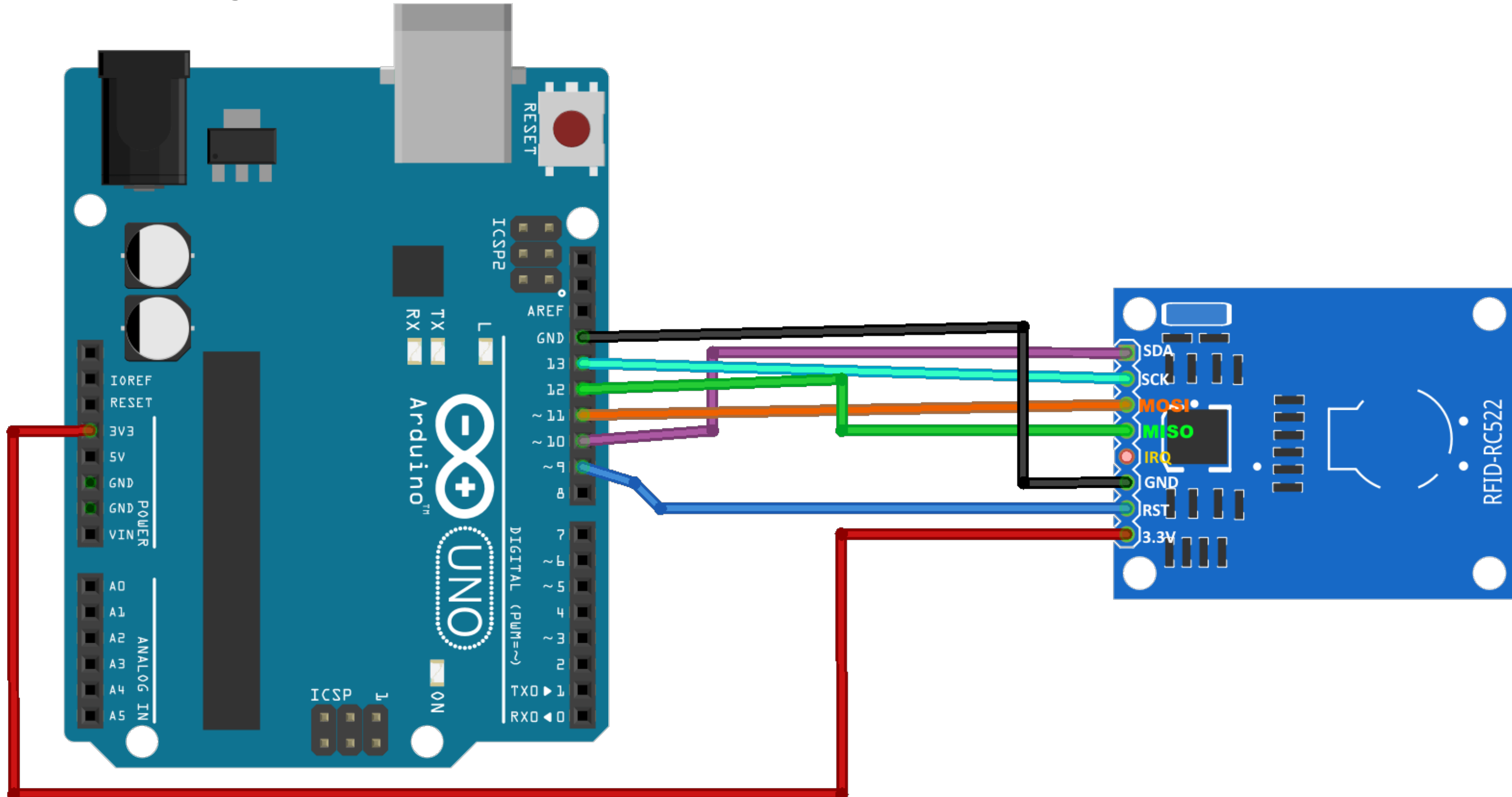


Reading Data From RFID Card

In this section we'll learn to read data saved on the RFID card.

This has an application in Tap&Go system where the passenger places the RFID card on the module located at the bus' entrance in order to get money deducted.

Connection Diagram:



Code:

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 9
#define SS_PIN 10
MFRC522 mfrc522(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;
MFRC522::StatusCode card_status;

void setup(){
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println(F("PCD Ready!"));
}

void loop(){
  for (byte i = 0; i < 6; i++){
    key.keyByte[i] = 0xFF;
  }
  if(!mfrc522.PICC_IsNewCardPresent()){
    return;
  }
  if(!mfrc522.PICC_ReadCardSerial()){
    return;
  }

  Serial.println(F("\n*** Balance on the PICC ***\n"));
  String balance = readBytesFromBlock();
  Serial.println(balance);
  Serial.println(F("\n*****\n"));
  delay(1000);

  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
}

String readBytesFromBlock(){
  byte blockNumber = 4;

  card_status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, blockNumber, &key, &(mfrc522.uid));
  if(card_status != MFRC522::STATUS_OK){
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(card_status));
    return;
  }

  byte arrayAddress[18];
  byte buffersize = sizeof(arrayAddress);
  card_status = mfrc522.MIFARE_Read(blockNumber, arrayAddress, &buffersize);
  if(card_status != MFRC522::STATUS_OK){
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(card_status));
    return;
  }

  String value = "";
  for (uint8_t i = 0; i < 16; i++){
    value += (char)arrayAddress[i];
  }
  value.trim();
  return value;
}
```

How the code works

Import MFRC522, the RFID reader library	#include <MFRC522.h>
Instantiate a MFRC522 reader object with slave select pin and reset pin	MFRC522 mfrc522(SS_PIN, RST_PIN);
Create a MIFARE_Key struct named 'key', which will hold the card information	MFRC522::MIFARE_Key key;
Declare the block number to hold data written to the PICC	byte block=4;
Initialize MFRC522 PCD (Card Reader)	mfrc522.PCD_Init();
Prepare the security key for the read and write functions whereby keyByte is defined in the "MIFARE_Key" 'struct' definition in the MFRC522.h library	for (byte i = 0; i < 6; i++) { key.keyByte[i] = 0xFF; }
Identify if the PICC is new or has already been used. And if a new PICC placed to RFID reader continue.	mfrc522.PICC_IsNewCardPresent()
Select one of the cards located in the proximity	mfrc522.PICC_ReadCardSerial()
Stop reading the card	mfrc522.PICC_HaltA();
Stop encryption on PCD	mfrc522.PCD_StopCrypto1();
We need to define a variable with the read buffer size, since the MIFARE_Read method needs a pointer to the variable that contains the size, &buffersize is a pointer to the buffersize variable; MIFARE_Read requires a pointer instead of just a number.	byte buffersize = 18;

What does "struct" mean for "MFRC522::MIFARE_Key key;"?

Is this a static property in the class MFRC522?

int vs byte

A data type tells what a variable can hold in programming. A variable is a name we give to a piece of data in our memory. In general, we should pick the minimum for our task.

Because the number of blocks varies between 0 and 63

MIFARE Classic has 1 kilobytes of Memory

The 1K memory of the Tag is organized in 16 sectors (from 0 to 15)

Each sector is further divided into 4 blocks (block 0 to 3).

Each block can store 16 bytes of data (from 0 to 15).

That surely tells us we have 16 sectors x 4 blocks x 16 bytes of data = 1024 bytes = 1K memory.
