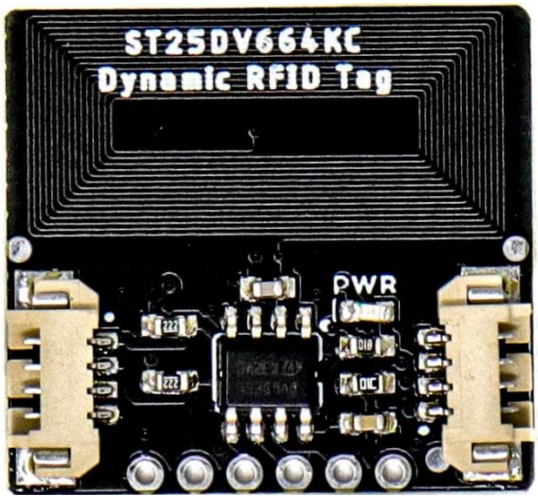
SmartElex Dynamic NFCRFID Tag

The Dynamic NFC/RFID Tag features the ST25DV64KC dynamic Near Frequency Communication (NFC) / Radio Frequency Identification (RFID) tag IC from STMicroelectronics©. The ST25DV64KC offers 64-kBit (8-kBytes) of EEPROM memory which can be accessed over both I2C and RF (NFC)! It's a state-of-the-art tag which conforms to ISO/IEC 15693 or NFC Forum Type 5 recommendations. You can read and write the tag's memory using NFC even while the tag is powered down or disconnected!



In this guide we'll go into some detail on the ST25DV64KC IC and other hardware on this breakout, how to assemble it into a circuit and then install the SparkFun ST25DV64KC Arduino Library.

ST25DV64KC Dynamic NFC/RFID Tag

The ST25DV64KC from STMicroelectronics is a unique tag IC that communicates over both I^2C and RF (NFC). It conforms to ISO/IEC 15693 (13.56 MHz frequency) and NFC Forum Type 5 recommendations.

The ST25DV64KC includes 64 Kbit EEPROM for users to write and read data from, a general purpose output to act as an external interrupt reporting events such as RF field changes, RF activity, I^2C writes and RF switch toggling over I^2C. The IC has a supply voltage range of **1.8V** to **5.5V** though when in a circuit it runs at **3.3V**. It also includes an energy harvesting pin capable of outputing µW of power with an RF field of sufficient strength. For a complete overview of the ST25DV64KC, refer to the datasheet.

The ST25DV64KC supports a fast transfer mode to send the contents of a 256 byte buffer between a device connected to the tag over I^2C (refered to as the tag's Mailbox) and an RF device such as a reader or smartphone. This makes it so you can store data on the tag and have it available for reading by an RF device by simply bringing it into the RF read range, even if the tag is powered off. This data can also be password protected with a 64-bit value.

Power and Communication Interfaces

The Dynamic NFC/RFID Tag has two interfaces for powering and communicating with the ST25DV64KC: a pair of connectors and a plated through hole (PTH) header.

Plated Through Hole Header

The board has a 0.1"-spaced PTH header that breaks out the power pins (3.3V and Ground), I^2C interface (SDA/SCL), energy harvesting pin (VEH), and general purpose output pin (GPO).

The Energy Harvesting Pin outputs an analog voltage when energy harvesting mode is enabled and in the presence of a strong enough RF field. Refer to section 5.5 of the datasheet for specifics on using this pin. The General Purpose Output is an open-drain configurable pin used for interrupt events.

Antenna

The board includes a PCB antenna for the ST25DV64KC to help boost the read range a bit. In our testing with a smart phone running an NFC reader app we found it had a range of a few centimeters.

LED

The sole LED on this board is a power status LED.

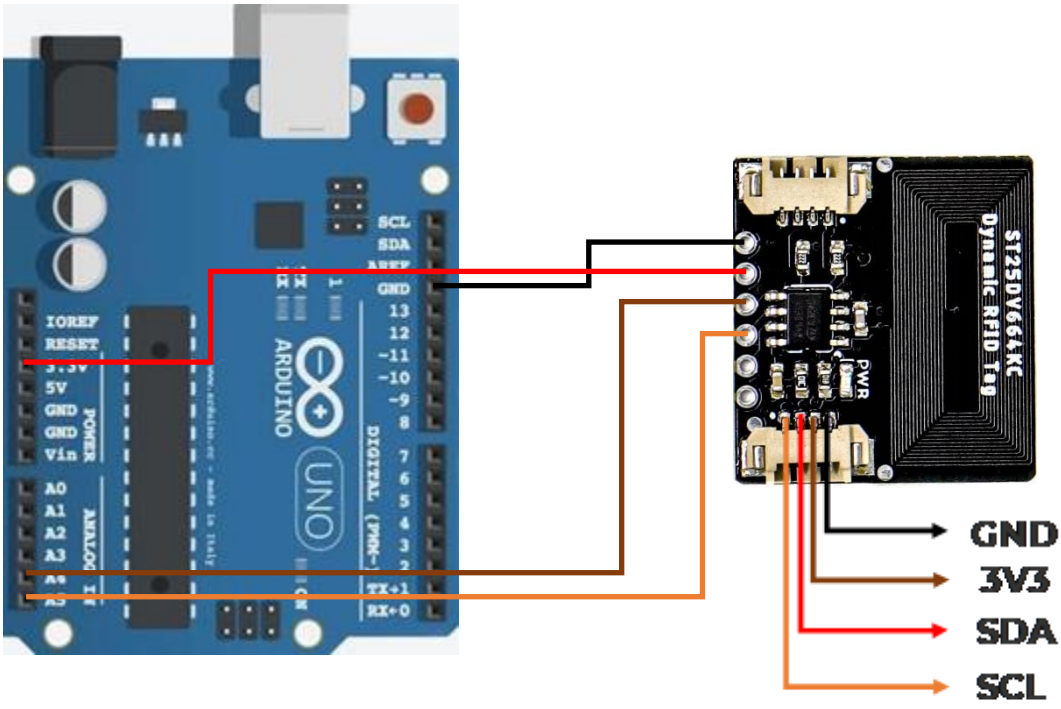
Solder Jumpers

The board has three solder jumpers labeled **LED**, **GPO** and **I2C**. The table below outlines their labels, default state, functionality, and any notes regarding their use.

Label	Default State	Function	Notes
LED	CLOSED	Completes the Power LED circuit.	Open to disable the Power LED.
GPO	CLOSED	Pulls the interrupt (GPO) pin HIGH through a **10kΩ** resistor.	Open to disable the pullup resistor.
I2C	CLOSED	Pulls the SDA/SCL lines to VCC (**3.3V**) through a pair of **2.2kΩ** resistors.	Open to disable the pullup resistors.

As a reminder, the connectors only include the power pins and I^2C lines so if you wish to use the Energy Harvesting pin or the General Purpose pin you'll need to solder to them or use something like these IC Hooks for a temporary prototyping connection.

Wiring:



Arduino	NFC/RFID
A5(SCL)	SCL
A4(SDA)	SDA
3.3V	3V3
GND	GND

SparkFun ST25DV64KC Arduino Library

The SparkFun ST25DV64KC Arduino Library provides an exhaustive collection of examples to take full advantage of the Dynamic NFC/RFID Tag including reading and writing the user memory, controlling the read/write permissions, altering the read area sizes, and applying password control. On top of this the library includes examples showing how to use NDEF (NFC Forum Data Exchange Format) URI, WiFi, and Text records you can read with a smart phone. You'll need a compatible app on your phone like ST's "NFC Tap" App available on the Apple© App store or Google© Play.

Install the library into Arduino by searching for "SparkFun ST25DV64KX Arduino Library" in the Arduino Library Manager.

Example Code: Read ID

```
#include <SparkFun_ST25DV64KC_Arduino_Library.h> // Click here to get the
library:  http://librarymanager/All#SparkFun_ST25DV64KC

SFE_ST25DV64KC tag;

void setup()
{
  delay(1000);

  Serial.begin(115200);
  Wire.begin();

  Serial.println(F("ST25DV64KC example."));
```

```
  if (!tag.begin(Wire))
  {
    Serial.println(F("ST25 not detected. Freezing..."));
    while (1) // Do nothing more
      ;
  }

  Serial.println(F("ST25 connected."));

  uint8_t values[8] = {0};
  if (tag.getDeviceUID(values))
  {
    Serial.print(F("Device UID: "));
    for (uint8_t i = 0; i < 8; i++)
    {
      if (values[i] < 0x0a)
        Serial.print(F("0"));
      Serial.print(values[i], HEX);
      Serial.print(F(" "));
    }
    Serial.println();
  }
  else
    Serial.println(F("Could not read device UID!"));

  uint8_t rev;
  if (tag.getDeviceRevision(&rev))
  {
    Serial.print(F("Revision: "));
    Serial.println(rev);
  }
  else
    Serial.println(F("Could not read device revision!"));
}

void loop()
{
  // Nothing to do here
}
//////////////////////////////////////////END/////////////////////////////////////////
```