Babel Fish

Created by Becky Stern

https://learn.adafruit.com/babel-fish

Last updated on 2023-08-29 02:10:30 PM EDT

Table of Contents

Overview 3

Make the NFC/RFID Shield 3

Make the Wave Shield 5

Flash cards 5

Programming 6

Fish Box 9

Use it! 11

Overview

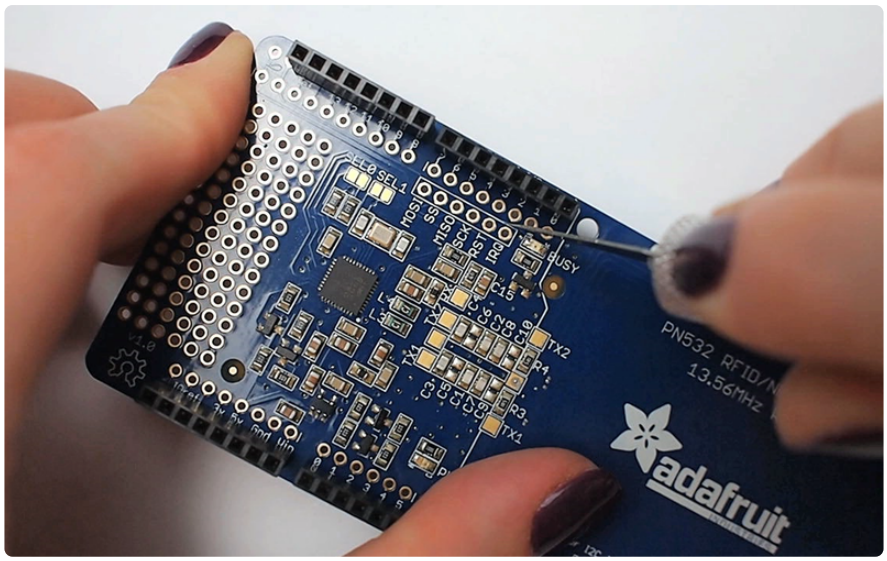
Learn to make a speaking, card-reading toy! The Babel Fish helps you learn to say words and phrases on RFID flash cards by reading the card and playing an associated sound file loaded on an SD card inside. This project is very straightforward and could make a great jumping-off point for your own awesome RFID and Wave Shield project!

For this project you will need

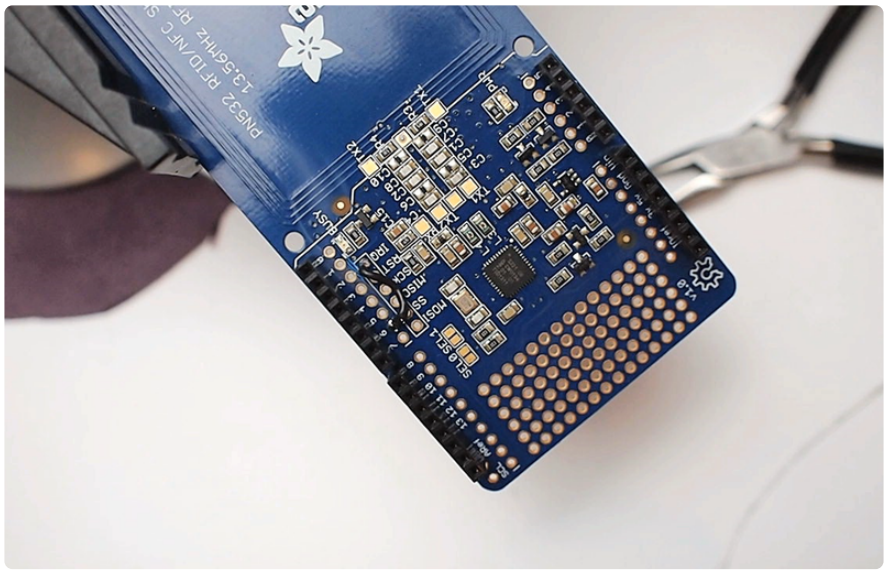
- Arduino Uno ()
- Adafruit NFC/RFID Shield ()
- Adafruit Wave Shield () (and your own SD card & speaker) or Wave shield starter kit ()
- MiFare cards (one per flashcard) ()
- Power adapter ()

Make the NFC/RFID Shield

Watch our video about the NFC/RFID shield for Arduino ()! We need to modify the shield to work with the Wave Shield.

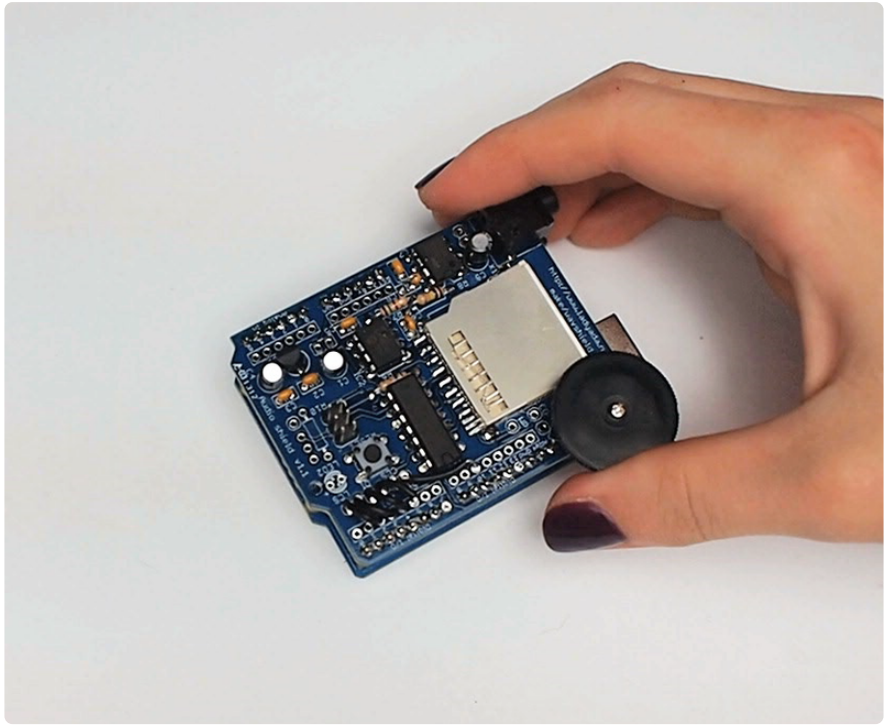


Use a utility knife to cut the connection between the IRQ pin and header pin neighbor.



Solder a wire from the IRQ pin to digital pin 6.

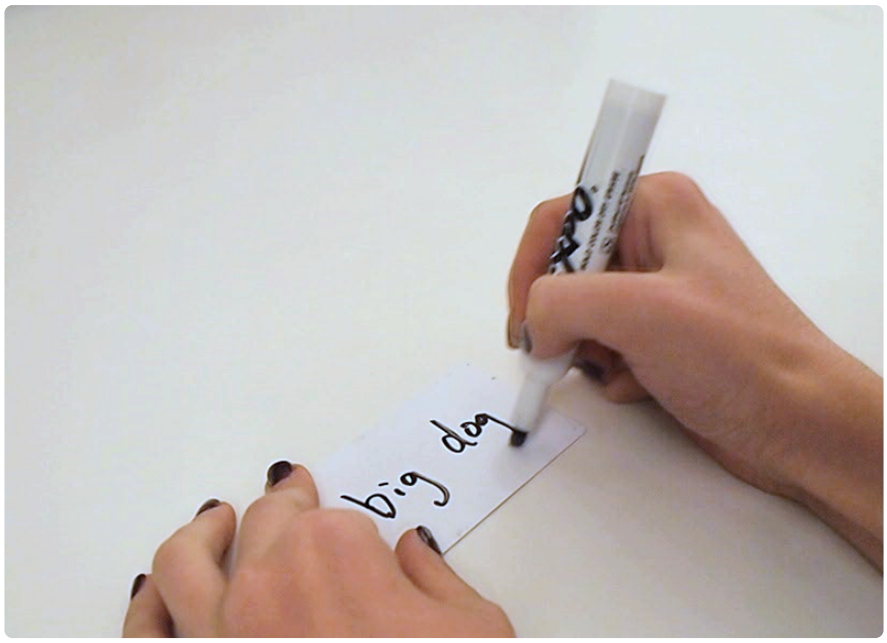
Make the Wave Shield



Refer to the Wave Shield tutorial () for complete instructions for assembly. Use the instructed wiring like in the sample. Test the shield to be sure it works before combining with another shield.

Flash cards

You can make flash cards by printing our words and phrases and taping them to RFID cards.



You can also just write on the RFID cards with a dry-erase marker.

Programming

Program your Arduino with the following sketch and open the serial monitor.

Dont forget you'll need to have the Waveshield WaveHC library, and NFCshield library installed first. Visit the Wave and NFC shield product pages and test both of them before continuing onto this code!

```
#include &lt;WaveHC.h&gt;
#include &lt;WaveUtil.h&gt;
#include &lt;Wire.h&gt;
#include &lt;Adafruit_NFCShield_I2C.h&gt;


#define IRQ 6 // this trace must be cut and rewired!
#define RESET 8

Adafruit_NFCShield_I2C nfc(IRQ, RESET);

SdReader card; // This object holds the information for the card
FatVolume vol; // This holds the information for the partition on the card
FatReader root; // This holds the information for the volumes root directory
FatReader file; // This object represent the WAV file for a pi digit or period
WaveHC wave; // This is the only wave (audio) object, since we will only play one
at a time
/*
* Define macro to put error messages in flash memory
*/
#define error(msg) error_P(PSTR(msg))

////////////////////////////////// SETUP

void setup() {
  // set up Serial library at 9600 bps
  Serial.begin(9600);
```

```
  PgmPrintln("Pi speaker");

  if (!card.init()) {
    error("Card init. failed!");
  }
  if (!vol.init(card)) {
    error("No partition!");
  }
  if (!root.openRoot(vol)) {
    error("Couldn't open dir");
  }

  PgmPrintln("Files found:");
  root.ls();

  // find Adafruit RFID/NFC shield
  nfc.begin();

  uint32_t versiondata = nfc.getFirmwareVersion();
  if (! versiondata) {
    Serial.print("Didn't find PN53x board");
    while (1); // halt
  }
  // Got ok data, print it out!
  Serial.print("Found chip PN5"); Serial.println((versiondata&gt;&gt;24) &amp;
0xFF, HEX);
  Serial.print("Firmware ver. "); Serial.print((versiondata&gt;&gt;16) &amp; 0xFF,
DEC);
  Serial.print('.'); Serial.println((versiondata&gt;&gt;8) &amp; 0xFF, DEC);

  // configure board to read RFID tags
  nfc.SAMConfig();

}

/////////////////////////////////// LOOP

unsigned digit = 0;

void loop() {
  uint8_t success;
  uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 }; // Buffer to store the returned UID
  uint8_t uidLength; // Length of the UID (4 or 7 bytes depending on ISO14443A card
type)

  // wait for RFID card to show up!
  Serial.println("Waiting for an ISO14443A Card ...");


  // Wait for an ISO14443A type cards (Mifare, etc.). When one is found
  // 'uid' will be populated with the UID, and uidLength will indicate
  // if the uid is 4 bytes (Mifare Classic) or 7 bytes (Mifare Ultralight)
  success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &amp;uidLength);

  uint32_t cardidentifier = 0;

  if (success) {
    // Found a card!

    Serial.print("Card detected #");
    // turn the four byte UID of a mifare classic into a single variable #
    cardidentifier = uid[3];
    cardidentifier &lt;&lt;= 8; cardidentifier |= uid[2];
    cardidentifier &lt;&lt;= 8; cardidentifier |= uid[1];
    cardidentifier &lt;&lt;= 8; cardidentifier |= uid[0];
    Serial.println(cardidentifier);

  // repeat this for loop as many times as you have RFID cards
    if (cardidentifier == 2588581390) { // this is the card's unique identifier
```

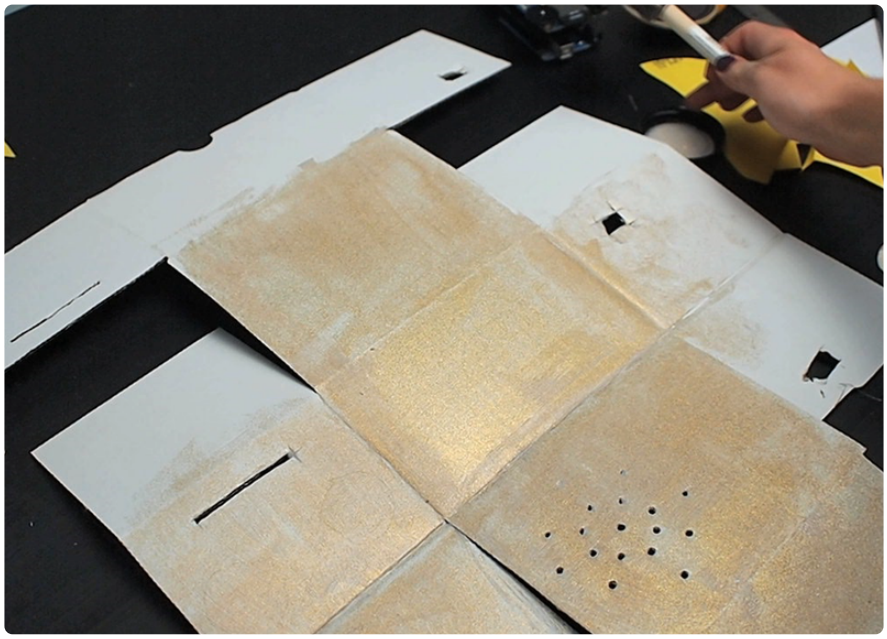
```
      playcomplete("1.WAV"); // these are file names for the sample audio files -
change them to your own file names
    }

    if (cardidentifier == 2146122274) {
      playcomplete("2.WAV");
    }
  }
}

////////////////////////////////// HELPERS

/*
* print error message and halt
*/
void error_P(const char *str) {
  PgmPrint("Error: ");
  SerialPrint_P(str);
  sdErrorCheck();
  while(1);
}
/*
* print error message and halt if SD I/O error
*/
void sdErrorCheck(void) {
  if (!card.errorCode()) return;
  PgmPrint("\r\nSD I/O error: ");
  Serial.print(card.errorCode(), HEX);
  PgmPrint(", ");
  Serial.println(card.errorData(), HEX);
  while(1);
}
/*
* Play a file and wait for it to complete
*/
void playcomplete(char *name) {
  playfile(name);
  while (wave.isplaying);

  // see if an error occurred while playing
  sdErrorCheck();
}
/*
* Open and start playing a WAV file
*/
void playfile(char *name) {
  if (wave.isplaying) {// already playing something, so stop it!
    wave.stop(); // stop it
  }
  if (!file.open(root, name)) {
    PgmPrint("Couldn't open file ");
    Serial.print(name);
    return;
  }
  if (!wave.create(file)) {
    PgmPrintln("Not a valid WAV");
    return;
  }
  // ok time to play!
  wave.play();
}
```

Replace the long number in the following for loop with your RFID card's ID. Replace the sound file name with the filename for your chosen sound samples.

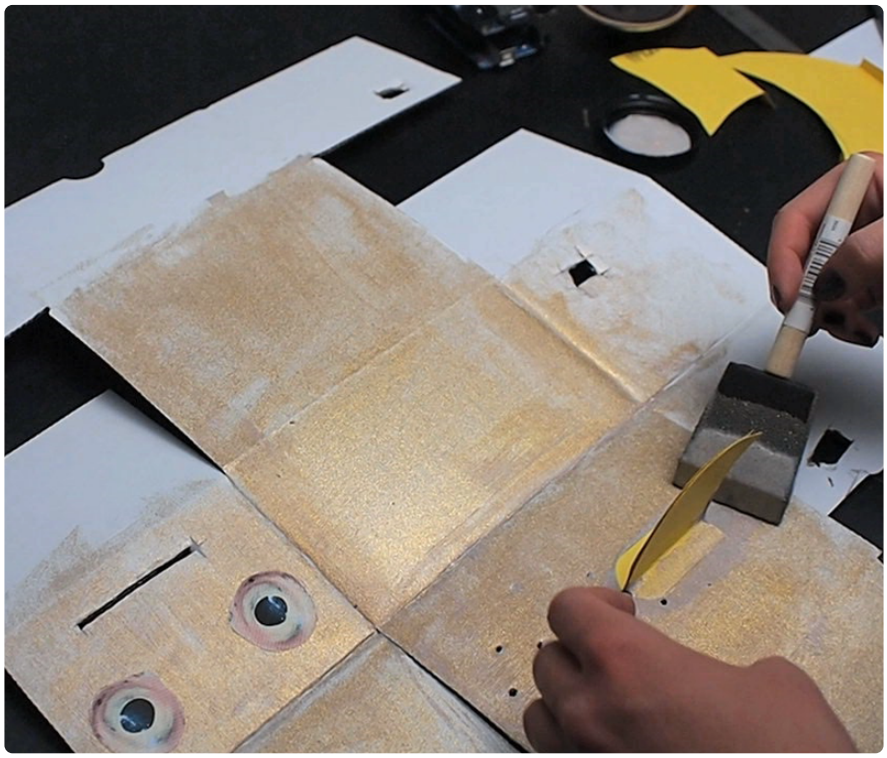
```
  if (cardidentifier == 2146122274) {
    playcomplete("2.WAV");
  }
```

Fish Box

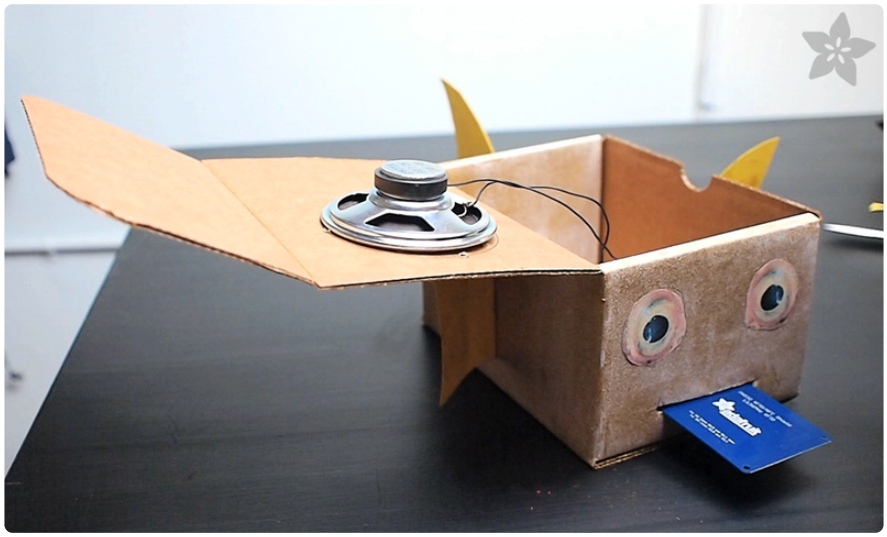
Mark and cut the box with holes for the RFID antenna, speaker, and power cable.



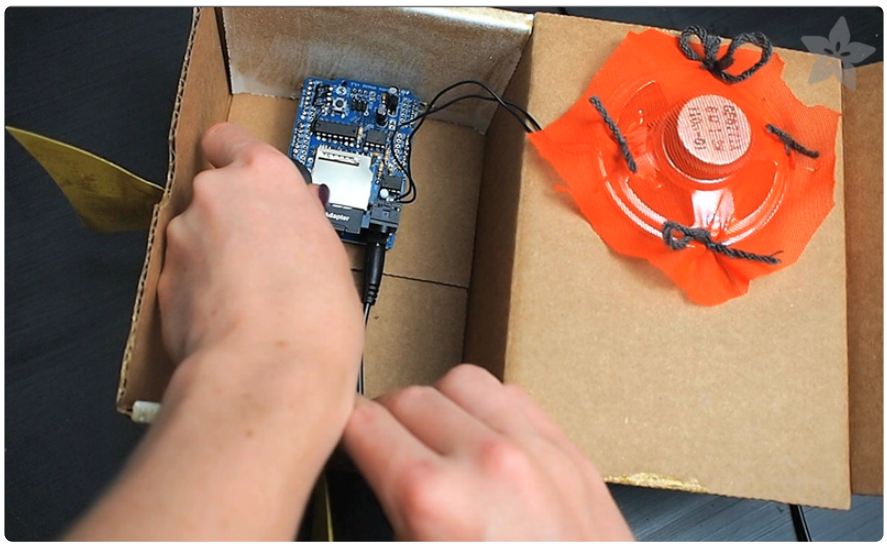
Coat a plain cardboard box in glitter Mod Podge.



The Mod Podge also works as an adhesive! Use it to attach eyes (google image search) and fins (yellow file folder).



Once dry, assemble the box and install the electronics.



Using a stretchy scrap of fabric, stitch the speaker in place with yarn and a yarn needle. This is a good temporary speaker mount you can easily undo later and is recommended over hot glue!

Use it!

You can reprogram your sketch for new words and phrases any time!