

Boek 10: Zender/Ontvanger

Inhoudsopgave:

1. De Zender:	2
2. De Ontvanger:	3
3. Een library installeren.	3
4. Arduino code voor de zender (Tx):	4
5. Arduino code voor de ontvanger:	5



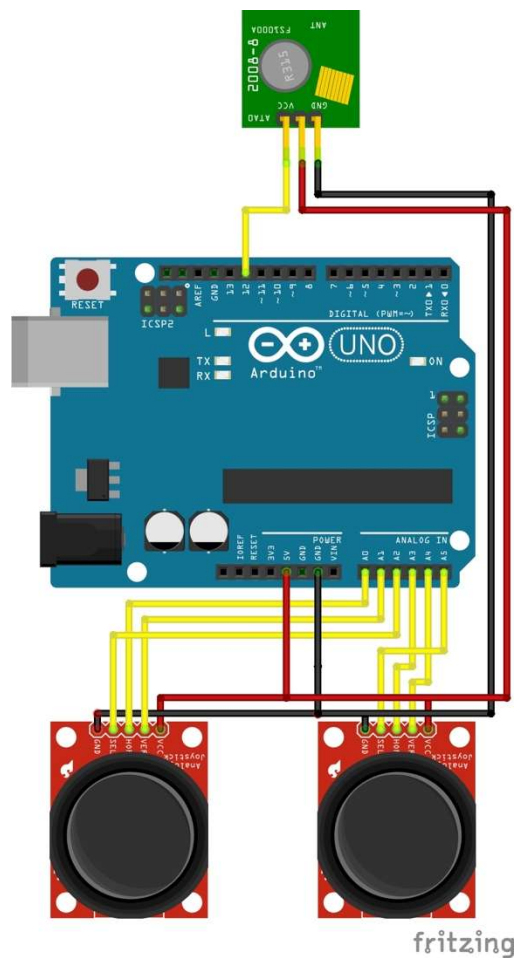
(C) Arduino cursus Groningen 2023

1. De Zender:

De zender en ontvanger in deze les werken op 433MHz (Mega Hertz). Dit is een radio frequentie die gebruikt wordt voor allerlei afstandsbedieningen en portofoons (walkie talkies). Deze frequentie is vrij van vergunning en kan dus door iedereen gebruikt worden.

De zender zet de digitale bitjes die op pin 12 van de Arduino worden verzonden om in kleine signalen die via de 433MHz worden uitgezonden. Door gebruik te maken van een library die speciaal voor deze modules is gemaakt is het eenvoudig om data van de zender naar de ontvanger te sturen. Het is eenrichtingsverkeer, dus geen terugmelding van de zender naar ontvanger.

De data die we versturen stoppen we in een "Data_Package" zodat we van tevoren aangeven hoeveel bytes en in welke volgorde deze worden verstuurd. Hierdoor weet de ontvanger of alle data ontvangen is.



Figuur 1: De zender.

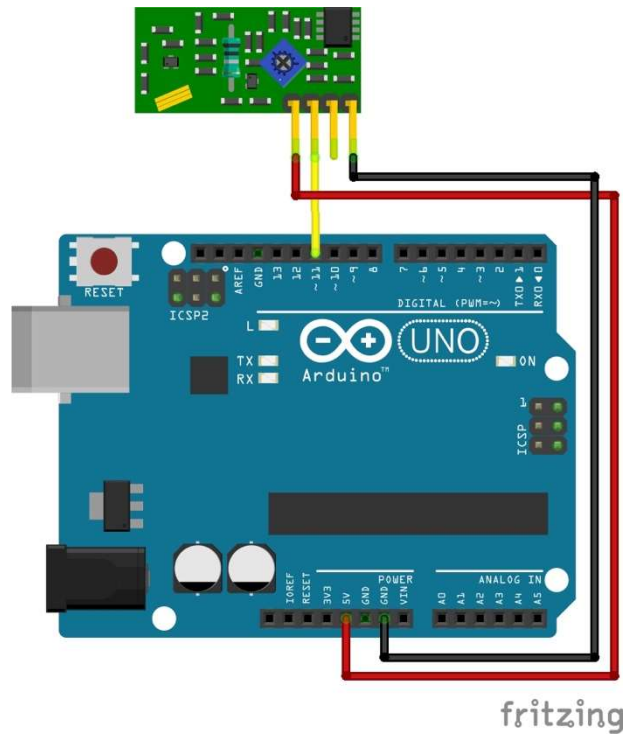
De zender in dit voorbeeld gebruikt 2 joysticks met elk ook een druktoets. Eventueel zijn er extra bronnen aan te sluiten en die data draadloos over te sturen. Hiervoor moeten de extra datasignalen aan het "data_package" van de zender worden toegevoegd. Ook aan de ontvangerzijde moet deze extra worden toegevoegd zodat de zender en ontvanger over dezelfde inhoud van het pakket beschikken.

De verzonden commando's in deze les zijn:

Joystick1 X	Joystick2 X
Joystick1 Y	Joystick2 Y
Joystick1 drukknop	Joystick2 drukknop

2. De Ontvanger:

De ontvangst kant is een eenvoudige schakeling waarbij de ontvanger met 3 draden op de Arduino wordt aangesloten. Een +5V, Gnd en een data lijn die de ontvangen data naar pin 11 laat gaan.



3. Een library installeren.

Voor de zender en ontvanger is een aparte library nodig die alle software afhandelt voor de communicatie. Hieronder staat hoe je een "library" oftewel een bibliotheek met extra commando's en aansturing moet installeren.

De volgende libraries gebruiken we:

```
#include <RH_ASK.h>           // Zender/Ontvanger module aansturing
#include <SPI.h>               // Communicatie
```

De library kan handmatig vanaf een ZIP file worden geïnstalleerd. Hiervoor is op de Google drive de file toegevoegd. Via: Sketch, Include Library, Add "RadioHead-master.zip" Library.

Het is ook mogelijk de zip files uit te pakken en in C:\Documents\Arduino\Libraries\... te zetten en dan Arduino even opnieuw opstarten:

C:\Documents\Arduino\Libraries\RadioHead

4. Arduino code voor de zender (Tx):

```
/*
*****
* AW Arduino Zender code (Tx).      versie 06-05-2023 *
*
* Arduino      Transmitter *
* GND-----GND *
* D12----->>-----Data *
* 5V-----VCC *
*****
#include <RH_ASK.h>           // Tx data komt uit pin 12
#include <SPI.h>

RH_ASK rf_driver(2000, 11, 12); // 2000bps, Rx op pin 11, Tx op pin 12

struct Data_Package{         // de data heeft de volgende parameters:
  byte joy1X;                // analoge data van 0-1023 omgezet in 0-255 (byte)
  byte joy1Y;
  byte joy1S;
  byte joy2X;
  byte joy2Y;
  byte joy2S;
};

Data_Package data;          // Maak het data pakket

void setup(){
  Serial.begin(115200);
  rf_driver.init();         // start de draadloze_verbindings driver
  Serial.println("starting transmitter");
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT_PULLUP);
  pinMode(A3, INPUT);
  pinMode(A4, INPUT);
  pinMode(A5, INPUT_PULLUP);
}

void loop(){
  data.joy1X = map(analogRead(A0), 0, 1023, 255, 0); // Maak een Byte van de analoge data
  data.joy1Y = map(analogRead(A1), 0, 1023, 0, 255); // en verwissel links en rechts (min/max)
  data.joy2X = map(analogRead(A3), 0, 1023, 255, 0); //
  data.joy2Y = map(analogRead(A4), 0, 1023, 0, 255); //
  data.joy1S = digitalRead(A2);
  data.joy2S = digitalRead(A5);
  Serial.print(data.joy1X); Serial.print(" ");
  Serial.print(data.joy1Y); Serial.print(" ");
  Serial.print(data.joy1S); Serial.print(" ");
  Serial.print(data.joy2X); Serial.print(" ");
  Serial.print(data.joy2Y); Serial.print(" ");
  Serial.println(data.joy2S);
  rf_driver.send((uint8_t*)&data, sizeof(data)); // Zend het data pakket weg
  rf_driver.waitPacketSent(); // Wacht tot het pakket is verzonden
}
```

5. Arduino code voor de ontvanger:

```
/*
*****
* AW Arduino Ontvanger code (Tx). versie 06-05-2023 *
*
* Arduino Receiver *
* GND-----GND *
* D11-----<<-----Data *
* 5V-----VCC *
*****
#include <RH_ASK.h> // Rx data gaat naar pin 11
#include <SPI.h>

RH_ASK rf_driver(2000, 11, 12); // 2000bps, Rx op pin 11, Tx op pin 12

struct Data_Package{ // de data heeft de volgende parameters:
  byte joy1X; // analoge data van 0-1023 omgezet in 0-255 (byte)
  byte joy1Y;
  byte joy1S;
  byte joy2X;
  byte joy2Y;
  byte joy2S;
};
Data_Package data; // Maak het data packet

void setup(){
  Serial.begin(115200);
  rf_driver.init(); // start de draadloze_verbindings driver
  Serial.println("starting receiver");
}

void loop(){
  uint8_t buflen = sizeof(data); // maak een ontvangst buffer met de lengte van het data pakket
  if (rf_driver.recv((uint8_t*)&data, &buflen) && buflen == sizeof(data)){ // is er een volledig pakket ontvangen ?
    Serial.print(data.joy1X); Serial.print(" "); // print de data van het pakket
    Serial.print(data.joy1Y); Serial.print(" ");
    Serial.print(data.joy1S); Serial.print(" ");
    Serial.print(data.joy2X); Serial.print(" ");
    Serial.print(data.joy2Y); Serial.print(" ");
    Serial.println(data.joy2S);
  }
}
```



Probeer nu zelf eens de andere mogelijkheden van de besturing !!
