ELECHOUSE_CC1101 Library Instruction

Interface Defination

- VCC: 3.3v, not more than 3. 6v
- GND: Ground
- SCLK: Clock, connecting with SCK of Arduino spi
- SI: Serial data input, connecting with MOSI of Arduino
- SO: serial data output, connecting with MISO of Arduino
- CSN: enable, connecting with SS of Arduino
- GDO0: output, as a symbol of sending or receiving data (refer to the register configuration of CC1101 datasheet). It is set to 1 while data packets start sending or receiving, and return to 0 while data sending or receiving is finished. Connected with Arduino digital pin 2(Defined in ELECHOUSE_CC1101.h) If you need to change it, modify the macro definition in ELECHOUSE_CC1101.h.
- GDO2: serial clock output (refer to register configuration of CC1101 datasheet), connecting with the Arduino digital pin 9 (Defined in ELECHOUSE_CC1101.h)

Function

Copy ELECHOUSE_CC1101 under \ libraries directory of IDE. Functions used to send or receive data as follows:

- void Init (void);
- void SendData (byte * txBuffer, byte size);
- void SetReceive (void);
- byte CheckReceiveFlag (void);
- byte ReceiveData (byte * rxBuffer);

1. void Init (void);

Initialize the CC1101module, which should be called in the **setup ()**;

```
void setup ()
{
ELECHOUSE_CC1101.Init ();
}
```

2. void SendData (byte * txBuffer, byte size);

Send data through CC1101 module. **txBuffer** array is the data, and **size** is the number of bytes to be sent. Pay attention that **size** should not be more than 61.

```
void loop ()
{
ELECHOUSE_CC1101.SendData (TX_buffer, size);
}
```

3. void SetReceive (void);

Set CC1101 module to state of expecting receiving data. It should be called in the **setup ()**. After receiving a packet of data, you need to call this function again if you want to receive more data. Call in the form of **ELECHOUSE_CC1101.SetReceive ()**;

4. byte CheckReceiveFlag (void);

Check whether data is received. This function is usually used if receiving data in the query mode (Another mode is interupt). Using this function, GDO0 connected with digital pin (default Pin2) by modifying the macro definition in **ELECHOUSE_CC1101.h**. It is not used if receiving data in interrupt mode, while GDO0 connects with an external interrupt pin.

```
if (ELECHOUSE_CC1101.CheckReceiveFlag ())
{
size = ELECHOUSE_CC1101.ReceiveData (RX_buffer);
ELECHOUSE_CC1101.SetReceive ();
}
```

5. byte ReceiveData (byte * rxBuffer);

Receive the data saved to **rxBuffer**, and return the number of bytes received. The maximum is 61 bytes.

```
{
size = ELECHOUSE_CC1101.ReceiveData (RX_buffer);
ELECHOUSE_CC1101.SetReceive ();
}
```

Example code:

1. Send 0-60

```
# Include <ELECHOUSE_CC1101.h>
# Define size 61
```

```
byte TX_buffer [size] = {0};
byte i;

void setup ()
{
  Serial.begin (9600);
  ELECHOUSE_CC1101.Init ();
  for (i = 0; i <size; i + +)
    {
        TX_buffer [i] = i;
    }
}

void loop ()
{
  ELECHOUSE_CC1101.SendData (TX_buffer, size);
  delay (1);
}
```

2. Receive data in query mode and display in the Serial Monitor

```
# Include <ELECHOUSE_CC1101.h>
void setup ()
{
  Serial.begin (9600);
  ELECHOUSE_CC1101.Init ();
  ELECHOUSE_CC1101.SetReceive ();
  }

byte RX_buffer [61] = {0};
byte size, i, flag;

void loop ()
{
  if (ELECHOUSE_CC1101.CheckReceiveFlag ())
  {
      size = ELECHOUSE_CC1101.ReceiveData (RX_buffer);
      for (i = 0; i <size; i + +)
      {
          Serial.print (RX_buffer [i], DEC);
          Serial.print (", BYTE);
      }
      Serial.println ("");
```

```
    ELECHOUSE_CC1101.SetReceive ();
  }
}
```

3. Receive data in interupt mode and display in the Serial Monitor

```
# Include <ELECHOUSE_CC1101.h>

void setup ()
{
  Serial.begin (9600);
  ELECHOUSE_CC1101.Init ();
  ELECHOUSE_CC1101.SetReceive ();
  attachInterrupt (0, ELECHOUSE_CC1101_RevData, FALLING);
}

byte RX_buffer [61] = {0};
byte size, i, flag;

void loop ()
{
  while (1);
}

void ELECHOUSE_CC1101_RevData ()
{
  size = ELECHOUSE_CC1101.ReceiveData (RX_buffer);
  for (i = 0; i <size; i + +)
  {
    Serial.print (RX_buffer [i], DEC);
    Serial.print ('', BYTE);
  }

Serial.println ("");
ELECHOUSE_CC1101.SetReceive ();
```