Library Version:	1.0
Library Reference:	https://github.com/rrobinet/RFM69OOK
Document Version:	0.0
Save Date:	21 June 2015

Introduction

The RFM69OOK Library is complementary to Moteino RFM69 Library (https://github.com/LowPowerLab/RFM69), it adds OOK (https://en.wikipedia.org/wiki/On-off_keying) transmission functionalities.
It allows an RFM69 transceivers working in FSK mode to be temporary send OOK commands to SAW (Surface Acoustic Wave) devices.
Most of these SAW devices are geared by protocols derived from X10 standard. (https://en.wikipedia.org/wiki/X10_(industry_standard)).
For Home automation one of the most popular (in The Netherlands and Belgium) is the KAKU (Klick Aan, Klick Uit). The current library covers several variants of this protocol, which are described under https://github.com/rrobinet/SAW_Devices_and_OOK/blob/master/OOK_Protocols_decription_V0.0.pdf.
There is no goal to develop a set of receive functions. Receive OOK frames with a RFM69 is complex and can't cohabitate easily with the main FSK modulation.

OOK datagrams can be decoded by other means, for more information, see:

- https://bitbucket.org/fuzzillogic/433mhzforarduino/wiki/Home.
- http://members.home.nl/hilcoklaassen/index.html
- https://github.com/SevenW/embapps/tree/master/ArduinoRFM69

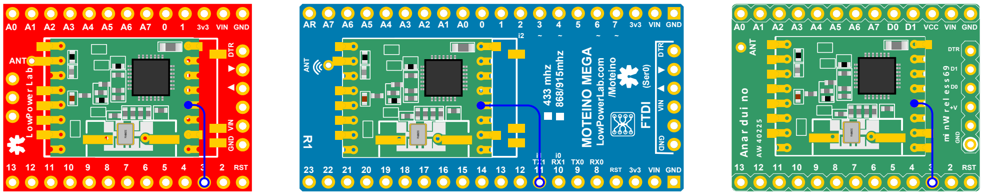
Description

OOK is an amplitude modulation of the transceiver carrier which in our case 433MHz.
By calling an OOK function, the native FSK modulation mode is momentarily suspended and the transceiver is entering the OOK mode for the duration of the function.
In OOK the RFM69 transmitter is operating in continuous mode. OOK modulation is obtained by switching on and off the power amplifier at the rate defined by the signal applied on the RFM69 transceiver DIO2 pin.

To operate the pin DI02 of transceiver, some hardware modification is necessary so that this pin of the RFM69 is connected to the processor.
For Moteino / Anarduino the proposed configuration is the following

Signal	Moteino pin	Moteino Mega pin	Anarduino MiniWireless69 pin
DIO2	3	11	3

Table 1: MOTEINO RFM69 DIO2 pin configuration



Moteino R4 Moteino Mega miniWireless69

Figure 1: Moteino, Moteino MEGA and MiniWireless69 DIO2 connection

Limitations

- Used only for OOK TRANSMISSION with RFM69(H)W transceivers at 433MHz
- Functions are limited to a set of well-known OOK protocols aka KAKU (the ones that I could test!)
  More specifically to KAKU OLD, KAKU NEW and KAKU GOGEX devices using the following remote control handsets
  - CHACON Handset Type: LYCT-505 (New Kaku)
  - CHACON Wireless Timer Control Handset Type: TMT-502 (Old Kaku)
  - COGEX Remote command Handset Type: 91789
- The functions are made to clone (replay) commands to **configured** devices, it is therefore important the know the parameters of the configured SAW devices to manage before using these functions, i.e.:

- One "House code" of one handset (New or Old) Ex: New: 1332798 or Old: D or Cogex: 12
- The "Unit address" of each device to manage
- The dimmable capability of the unit

RFM variables and constants

Most of the RFM variables and constant are the ones of the included RFM69 Library.

External variables

Boolean RFM69OOK_DEBUG;

- Debug option defined by the sketch, this one may be turned on or off by invoking in the sketch boolean RFM69OOK_DEBUG = true; or boolean RFM69OOK_DEBUG = false;

Configuration variables (private)

- RF69_OOK_PIN
  Processor OOK data pin connected to the RM60 DIO2 pin according to the processor type
  The default is:
  ATmega328 pin D3
  ATmega1284P pin D11
  ATmega2560 pin 11
  ATmega32U4 pin D2
- _ookDataPin
  The actual processor pin (default is RF69_OOK_PIN)
- _periodusec
  The average OOK pulse time for OOK SAW (default is 300µs)
- _repeats
  Number of time the OOK datagram is repeated (default is 10)
- repDly
  Delay between datagrams (default is 20ms)

RFM69OOK Class

Syntax

RFMOOK
RFM69OOK.setOokPin
RFM69OOK.setOokParams
RFM69OOK.sendKakuNew
RFMOOK69.sendKakuOld
RFMOOK69.sendKakuCogex

RFM69OOK

Description

A call to RFM69OOK creates a RFM69OOK instance, whose name needs to be provided while calling the class.

Syntax

RFM69OOK;
RFM69OOK (byte dataPin, unsigned int periodusec, byte repeats, byte repDly);

Notes

Multiple instances may be created however they will all inherit of the same variables, it is therefore better to change the parameters of one defined instance using setOokParams instead of created several instances.

Parameters

none:
Default *dataPin* is RF69_OOK_PIN, the default *periodusec* is 300µs, default *repeats* is 10, the default *repDly* is 20ms.

- *dataPin:*
  Used to select a processor pin connected to the RFM69 DIO2 pin
  See Table 1: MOTEINO RFM69 DIO2 pin configuration on page 1.
- *periodusec:*
  Basic OOK symbol pulse period in µs depending of the OOK specific protocol
- *repeats:*
  Number of time the OOK datagram is to be repeated

- *repDly:*
  Delay between each repeated datagram

Example

```
RFM69OOK.switchKaku;
RFM69OOK.switchKakuCogex(3,350,15,15);  // Setup one RFM69OOK instance with Data Out on pin 3, symbol period of 350us, number of
repeated datagram 15 with an interval delay of 20ms
```

RFM69OOK.setOokPin

Description

This function allows changing the processor pin connected to the RFM69 DIO2 transceiver pin. It is a common parameter used by each RFM69OOK instance.

Syntax

RFM69OOK. setOokPin(uint8_t newOokDataPin)

Parameters

- *newOokDataPin Returns*
  An alternate processor pin connected to the RFM69 DIO2 pin

Returns

None

Example

```
RFM69OOK.switchKaku;
switchKaku.setOokPin (4);     // set the processor pin connected to RFM69 DIO2 to 4 t
```

RFM69.setOokParams

Description

This function allows changing the particular protocol parameters such as the symbol period, the delay between datagrams and the number of datagrams to be repeated.

Note

This function may be issued before each send request to a SAW device.

Syntax

RFM69OOK.setOokParams(unsigned int periodusec, byte repeats, byte repDly);

Parameters

- *periodusec:*
  The specific symbol period in µs
- *repeats:*
  The number of time the datagram is to be repeated
- *repDly:*
  The delay between repeated datagrams in ms

Returns

None

Example

```
switchKaku.setOokParams(375,10,11);     // Modify the OOK parameters for the kaku old send command
switchKaku.sendKakuOld(radio, 'D', 16, 1);
delay (5000);
switchKaku.setOokParams(350,10,20);     // Modify the OOK parameters for the kaku cogex send command
switchKaku.sendKakuCogex(radio, 12, 1, 1);
```

RFM69. sendKakuNew

Description

This function is used to send a command (On, Off, Dim level) from a programmed handset house address to a SAW device using the new Kaku protocol.

Syntax

RFM69OOK.sendKakuNew(RFM69 &radio, unsigned long int addr, byte unit, boolean on, boolean group, byte dimLevel);

Parameters

- *&radio:*
  The RFM69 instance associated to the target RFM69 OOK command

- *addr:*
  The House code of one active remote command handset in the new Kaku format (value comprise 1 to 67.108.86).
- *unit:*
  The actual switch unit to manage (from 1 to 16)
- *on:*
  A Boolean On / OFF level to be applied to the remote SAW switch
- *group:*
  A Boolean value that indicates that this command is for a set of switches
  The actual Unit address is not relevant when this parameter is used
- *dimlevel*:
  The dim level value to be applied to a particular SAW switch (that supports the dim function); the value may be comprise between 1 and 15

Returns

None

Example

```
switchKaku.sendKakuNew(radio,1332798,5,1,0,0); // Send to the RFM69 radio instance a command from house handset
1332798 to set switch unit 5 to ON, not group nor dim value are required.
```

RFM69. sendKakuOld

Description

This function is used to send a command (On, Off) from a programmed handset house code to a SAW device using the old Kaku protocol.

Syntax

RFM69OOK.sendKakuOld(RFM69 &radio, char addr, byte unit, boolean on)

Parameters

- *&radio:*
  The RFM69 instance associated to the target RFM69 OOK command
- *addr:*
  The House address of one active remote command handset n the old Kaku format (character comprise from 'A' to 'P').
- *unit:*
  The actual switch unit to manage (from 1 to 16)
- *on:*
  A Boolean On / OFF level to be applied to the remote SAW switch

Returns

None

Example

```
switchKaku.sendKakuOld(radio,'D',3,0); // Send to the RFM69 radio instance a command from house handset 'D' to
set switch unit 3 to OFF
```

RFM69. sendKakuCogex

Description

This function is used to send a command (On, Off) from a programmed handset house address to a SAW device using the Cogex Kaku protocol.

Syntax

RFM69OOK.sendKakuCogex(RFM69 &radio, byte addr, byte unit, boolean on)

Parameters

- *&radio:*
  The RFM69 instance associated to the target RFM69 OOK command
- *addr:*
  The House address of one active remote command handset in the Cogex Kaku format (byte comprise between 1 to 12).
- *unit:*
  The actual switch unit to manage (from 1 to 16)

- *on:*
  A Boolean On / OFF level to be applied to the remote SAW switch

Returns

None

Example

```
switchKaku.sendKakuCogex(radio,12,2,0); // Send to the RFM69 radio instance a command from house handset 12 to
set switch unit 2 to OFF
```

- ooOoo -