Arduino interface for LoRa Fabian

July 10th 2015

Contents

1 List of SPI messages

1.1 ARDUINO_CMD_AVAILABLE (0x00)

Get the value on the SPI.
On Available command, two bytes are returned to indicate the number of bytes available in the RX buffer.

1.2 ARDUINO_CMD_READ (0x01)

Send to the contiki a Read order to get the packet buffer.
On Read command the byte read is returned.

1.3 ARDUINO_CMD_WRITE (0x02)

Send a payload on the LoRa network.
This is the structure of a WRITE command packet:

0x02	length_msb	length_lsb	payload

1.4 ARDUINO_CMD_DEBUG (0x20)

Send a debug order to the shield.
This is the structure of a DEBUG command packet:

0x20	length_msb	length_lsb	0x00 (off) or 0x01 (on)

1.5 ARDUINO_CMD_HOSTNAME (0x21)

Send the name of the object to the shield.
This is the structure of a HOSTNAME command packet:

0x21	length_msb	length_lsb	hostname

1.6 ARDUINO_CMD_GET_MAC (0x22)

Ask the board to send its MAC. This is the structure of a GET_MAC command packet:

0x22

1.7 ARDUINO_CMD_FREQ (0x30)

Send the new frequency to the shield.
This is the structure of a FREQ command packet:

0x30	length_msb	length_lsb	freq (on 4 bytes)

1.8 ARDUINO_CMD_GET_FREQ (0x31)

Ask to write the frequency of the shield on the SPI. When the shield receives this order, it writes the frequency on four bytes length on the SPI. You can see the *getFreq()* function for an example.

1.9 ARDUINO_CMD_RF_CFG (0x32)

Send the new config to the shield.
This is the structure of a RF_CFG command packet:

0x32	length_msb	length_lsb	rfconfig

1.10 ARDUINO_CMD_BW_CFG (0x33)

Send the new bandwidth to the shield.
This is the structure of a BW_CFG command packet:

0x33	length_msb	length_lsb	bw

1.11 ARDUINO_CMD_GET_BW_CFG (0x34)

Ask to write the bandwidth of the shield on the SPI. When the shield receives this order, it writes the current bandwidth on one byte on the SPI. You can see the *getBandwidth()* function for an example.

1.12 ARDUINO_CMD_SF_CFG (0x35)

Send the new spreading factor to the shield.
This is the structure of a SF_CFG command packet:

0x35	length_msb	length_lsb	sf

1.13 ARDUINO_CMD_GET_SF_CFG (0x36)

Ask to write the spreading factor of the shield on the SPI. When the shield receives this order, it writes the current spreading factor on one byte on the SPI. You can see the *getSpreadingFactor()* function for an example.

1.14 ARDUINO_CMD_CR_CFG (0x37)

Send the new coding rate to the shield.
This is the structure of a CR_CFG command packet:

0x37	length_msb	length_lsb	bw

1.15 ARDUINO_CMD_GET_CR_CFG (0x38)

Ask to write the coding rate of the shield on the SPI. When the shield receives this order, it writes the current coding rate on one byte on the SPI. You can see the *getCodingRate()* function for an example.

2 LoRa Fabian Arduino Library

LoRaShield class functions:

2.1 LoraShield()

The constructor.

2.2 void init()

Initialize the SPI.

2.3 int dataAvailable()

Read the length of the packet available on the SPI.

2.4 void begin(String name)

- *Description:* This function tries to attach to a gateway from the LoRaFabian network.
- *Param: name* - The name of the node, example: "toto.s.ackl.io"

2.5 String read(bool verbose = false)

- *Description:* Read a packet on the SPI, check CRC and answer to a valid COAP request.
- *Param: verbose* - Print the payload and if the packet have a bad CRC.
- *Return:* The payload of the 802.15.4 packet received

2.6 void write(byte buff[], int bufflen)

- *Description:* Send a packet
- *Param: buff* - The payload
- *Param: bufflen* - The length of the buffer

2.7 void setContikiDebug(bool setcontikidebug)

- *Description:* Get all radio packets receives on the Contiki.
- *Param: setcontikidebug* - if true, the contiki will send all packets (including signalisation packets and packets for other destinations).

2.8 void setFreq(long freq)

- *Description:* Change the frequency of the radio
- *Param: freq* - the new frequency. Must be between 863000000 and 870000000.

Some macros:

- FREQ_8680: 868000000
- FREQ_8681: 868100000
- FREQ_8682: 868200000
- FREQ_8683: 868300000
- FREQ_8695: 869500000
- FREQ_8696: 869600000
- FREQ_8698: 869800000
- FREQ_8699: 869900000
- FREQ_MIN: 863000000
- FREQ_MAX: 870000000
- FREQ_DEFAULT: FREQ_MAX

2.9 unsigned long getFreq()

Read the current frequency from the contiki.

2.10 void setBandwidth(int bw)

- *Description:* Change the bandwidth of the radio
- *Param: bw* - the new bandwidth. Must be a following value : 0(125kHz), 1(250kHz), 2(500kHz)

Some macros:

- BW_MIN 0
- BW_MAX 2
- BW_DEFAULT 1

2.11 int getBandwidth()

Read the current bandwidth from the contiki.

2.12 void setCodingRate(int cr)

- *Description:* Change the coding rate of the radio
- *Param: cr* - the new coding rate. Must be a following value : 1(4/5), 2(4/6), 3(4/7), 4(4/8)

Some macros:

- CR_MIN 1
- CR_MAX 4
- CR_DEFAULT CR_MIN

2.13 int getCodingRate()

Read the current coding rate from the contiki.

2.14 void setSpreadingFactor(int sf)

- *Description:* Change the spreading factor of the radio
- *Param: sf* - the new spreading factor. Must be between 7 and 12

Some macros:

- SF_MIN 7
- SF_MAX 12
- SF_DEFAULT SF_MIN

2.15 int getSpreadingFactor()

Read the current spreading factor from the contiki board.

2.16 void setRFConfig(int rfconfig)

- *Description:* Change the whole configuration of the radio (spreading factor, bandwidth, coding rate)
- *Param: rfconfig* - the new configuration. Must be between 0 and 4.

Some macros:

- CONF_MIN 0
- CONF_MAX 4
- CONF_DEFAULT CONF_MIN

The configuration choice:

	SF	CR	BW
0	7	1	0
1	9	2	1
2	11	3	2
3	12	4	2
4	12	3	0

2.17 String getMAC()

Get a String object which contains the MAC address of the board.

3 Example

```
#include "LoraShield.h"
#include <SPI.h>

LoraShield lora;
void setup()
{
  Serial.begin(9600);
  while (!Serial); // wait for serial port to connect. Needed for Leonardo only

  Serial.println("Initialisation du Shield");
  lora.init();
  String host = "toto.s.ackl.io";
  Serial.println("Begin : " + host);
  lora.begin(host);
  Serial.println("Debug is off");
  lora.setContikiDebug(false);
  lora.setFreq(FREQ_8696);
  Serial.println("freq : " + String(lora.getFreq()));
```

```
  lora.setRFConfig(2);
}

void loop()
{
  delay(2000);
  int pktsize = lora.dataAvailable();
  if (pktsize > 0) {
    String msg = lora.read(true);
    if(msg != "") Serial.println("\nMessage: " + msg);
  }
  else Serial.println("0");
}
```